Melec

STEPPING & SERVO MOTOR CONTROLLER'S OPTION

MPL-34-02V1.00/AL2W32
MPL-35-02V1.00/AL2W64
取扱説明書 応用機能編 (設計者用)

(AL-IIシリーズ PCIマスター用Windowsデバイスドライバ)

MCC07ユニット編

USER'S MANUAL

既に本製品の別冊、取扱説明書を読まれていることを
前提に、より多彩な機能･仕様を解説した応用機能編です。
取扱説明書以上の内容についてはこの応用機能編を良く
読んで十分に理解してください。
この応用機能編は、いつでも取り出して読めるように
保管してください。

MN0279

はじめに

このデバイスドライバ「取扱説明書 応用機能編」は、AL-Ⅱ対応シリーズのステッピングモータ、サーボモータ、および I/O システムを正しく安全に使用していただくために、ステッピングモータ、あるいはサーボモータを使った制御装置の設計を担当される方を対象に、Windows における応用的な機能および仕様について説明しています。
各コントローラの「取扱説明書」ならびにデバイスドライバ「取扱説明書」と同様に、本デバイスドライバ「取扱説明書 応用機能編」を良く読んで十分に理解してください。
このデバイスドライバ「取扱説明書」は、いつでも取り出して読めるように保管してください。

安全設計に関するお願い

■本資料に記載される技術情報は、製品の代表的動作・応用を説明するためのものであり、その使用に際して当社および第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。

■本資料に記載されている回路、ソフトウェア、およびこれらに関連する情報を使用する場合は、お客様の機器およびシステム全体で十分に評価し、お客様の責任において適用可否を判断してください。

■半導体ならびに半導体を使用した製品は、ある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。本製品の故障または誤動作により、人身事故、火災事故、社会的な損害などを生じさせないように、お客様の責任において、お客様の機器またはシステムに必要な安全設計を行うことをお願いします。

■本製品は、一般工業向けの汎用品として設計・製造されていますので、航空機器、航空宇宙機器、海底中継機器、原子力制御システム、輸送機器(車両、船舶等)、交通用信号機器、防災・防犯機器、安全装置、医療機器など、人命や財産に多大な影響が予想される用途には使用しないでください。

■本製品を改造、改変、複製等しないでください。

■輸出に際しては、「外国為替および外国貿易法」など適用される輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続きを行ってください。本製品または本資料に記載されている技術情報を、大量破壊兵器の開発等の目的、軍事利用の目的、その他軍事用途の目的で使用しないでください。
また、本製品を国内外の法令および規制により製造・使用・販売を禁止されている機器に使用することはできません。

■本製品の環境適合性などの詳細につきましては、必ず弊社営業窓口までお問い合わせください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令など適用される環境関連法令を十分調査の上、かかる法令に適合するようにご使用ください。
お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は一切その責任は負いません。

安全に関する事項の記述方法について

本製品は正しい方法で取り扱うことが大切です。
誤った方法で使用された場合、予期しない事故を引き起こし、人身への障害や財産の損壊などの被害を被るおそれがあります。
そのような事故の多くは、危険な状況を予め知っていれば回避することができます。
そのため、このデバイスドライバ「取扱説明書」では危険な状況が予想できる場合には、注意事項が記述してあります。
それらの記述は、次のようなシンボルマークとシグナルワードで示しています。

⚠ 警告　取り扱いを誤った場合に死亡、または重傷を負うおそれのある警告事項を示します。

⚠ 注意　取り扱いを誤った場合に、軽傷を負うおそれや物的損害が発生するおそれがある注意事項を示します。

御使用の前に

■ AL-Ⅱ対応コントローラは各軸を独立で制御できるため、各軸を以下のように呼称します。
また、本書では、*1 の製品のことをコントローラドライバと呼称します。

製品名	1 軸目	2 軸目	3 軸目	4 軸目
2C-771v1	X 軸	Y 軸	Z 軸	A 軸
2C-776Av1	X 軸	Y 軸	Z 軸	A 軸
2CD-7710v1/ADB5F30 *1	X 軸	Y 軸	―	―
2CD-7713v1/GDB5F40 *1	X 軸	Y 軸	―	―

以降、原則として X 軸についてのみ説明します。

■ 入出力仕様ならびに接続に関する取り扱いについては、各コントローラの「取扱説明書」をご覧ください。

■ デバイスドライバを用いた基本的な仕様については、別冊デバイスドライバ「取扱説明書」をご覧ください。

はじめに
安全設計に関するお願い
安全に関する事項の記述方法について
御使用の前に

目　次

PAGE

1．概要
1-1. 特徴 ---- 7

2．関数リファレンス
2-1. MCC PORT アクセス関数の移行について ---- 8
2-2. デバイス関数 ---- 9
2-2-1. MCC PORT アクセス関数 ---- 9
DRIVE DATA 32 ビット書き込み関数 ---- 9
DRIVE DATA1 PORT 書き込み関数 ---- 10
DRIVE DATA2 PORT 書き込み関数 ---- 11
DRIVE DATA 32 ビット読み出し関数 ---- 12
DRIVE DATA1 PORT 読み出し関数 ---- 13
DRIVE DATA2 PORT 読み出し関数 ---- 14
データ構造体 ---- 15
DRIVE COMMAND データ構造体書き込み関数 ---- 16
DRIVE DATA データ構造体書き込み関数 ---- 17
DRIVE DATA データ構造体読み出し関数 ---- 18
データセット関数 ---- 19
データゲット関数 ---- 20
2-2-2. コマンド予約機能関数 ---- 21
COMREG NOT FULL WAIT 関数 ---- 21
DRIVE COMMAND バッファー書き込み関数 ---- 22
2-2-3. 円弧補間の演算関数 ---- 23
円弧補間短軸 PULSE 数ゲット関数 ---- 23
2-2-4. その他の関数 ---- 24
16 ビット符号なし変換関数 ---- 24
16 ビット符号付き変換関数 ---- 25
32 ビット符号なし変換関数 ---- 26
32 ビット符号付き変換関数 ---- 27

3．コマンド仕様
3-1. ドライブコマンド ---- 28
3-1-1. 入出力仕様の設定 ---- 28
(1) HARD INITIALIZE1 ---- 28
(2) HARD INITIALIZE4 ---- 30
(3) HARD INITIALIZE5 ---- 31
(4) HARD INITIALIZE6 ---- 32
(5) HARD INITIALIZE7 ---- 33
3-1-2. ドライブパラメータの設定 ---- 35
(1) FSPD SET ---- 35
(2) HIGH SPEED SET ---- 36
(3) LOW SPEED SET ---- 37
(4) RATE SET ---- 38
(5) SCAREA SET ---- 39
(6) DOWN PULSE ADJUST ---- 40
3-1-3. ORIGIN ドライブの設定 ---- 41
(1) ORIGIN SPEC SET ---- 41
(2) ORIGIN SCAN ---- 44
(3) ORIGIN CONSTANT SCAN ---- 45

目　次

PAGE

3-1-4. 任意軸補間ドライブの設定 ---- 46
(1) CP SPEC SET ---- 46
3-1-5. 直線補間ドライブの設定と実行 ---- 48
(1) LONG POSITION SET ---- 52
(2) SHORT POSITION SET ---- 53
(3) MAIN XY STRAIGHT CP ---- 54
(4) MAIN STRAIGHT CP ---- 55
(5) SUB STRAIGHT CP ---- 56
3-1-6. 円弧補間ドライブの設定と実行 ---- 57
(1) CIRCULAR XPOSITION SET ---- 61
(2) CIRCULAR YPOSITION SET ---- 62
(3) CIRCULAR PULSE SET ---- 63
(4) MAIN XY CIRCULAR CP ---- 64
(5) MAIN CIRCULAR CP ---- 65
(6) SUB CIRCULAR CP ---- 66
3-1-7. UP/DOWN/CONST ドライブ CHANGE の設定と実行 ---- 67
(1) UDC SPEC SET ---- 67
(2) UP DRIVE ---- 68
(3) DOWN DRIVE ---- 68
(4) CONST DRIVE ---- 68
3-1-8. SPEED CHANGE の設定と実行 ---- 69
(1) SPEED CHANGE SPEC SET ---- 69
(2) SPEED CHANGE ---- 70
3-1-9. RATE CHANGE の設定と実行 ---- 71
(1) RATE CHANGE ---- 71
3-1-10. INDEX CHANGE の設定と実行 ---- 72
(1) INDEX CHANGE SPEC SET ---- 72
(2) INC INDEX CHANGE ---- 73
(3) ABS INDEX CHANGE ---- 74
(4) PLS INDEX CHANGE ---- 75
3-2. カウンタコマンド ---- 76
3-2-1. アドレスカウンタの設定 ---- 76
(1) ADDRESS COUNTER MAX COUNT SET ---- 76
3-2-2. パルスカウンタの設定 ---- 77
(1) PULSE COUNTER MAX COUNT SET ---- 77
3-2-3. カウンタのラッチ・クリア機能の設定 ---- 78
(1) COUNT LATCH SPEC SET ---- 78
3-2-4. カウントデータのラッチデータの読み出し ---- 80
(1) ADDRESS LATCH DATA READ ---- 80
(2) PULSE LATCH DATA READ ---- 80
(3) DFL LATCH DATA READ ---- 80

３．機能説明

4-1. ドライブ仕様 ---- 81
4-1-1. コマンド予約機能 ---- 81
4-1-2. 入出力仕様 ---- 83
(1) 入力信号のデジタルフィルタ機能 ---- 83
(2) 入(出)力信号の論理切り替え機能 ---- 84
4-1-3. ドライブパラメータ ---- 85
(1) 加減速パラメータ ---- 85
(2) 加減速時定数 ---- 86
(3) 直線加減速ドライブ ---- 88
(4) S 字加減速ドライブ ---- 89
(5) 加速ドライブ ---- 93
(6) 減速ドライブ ---- 93
(7) 一定速ドライブ ---- 93
(8) その他のドライブ ---- 94

目　　次

PAGE

4-1-4. ORIGIN ドライブ (MCC07 チップの機械原点検出機能) ---- 95
4-1-5. 補間ドライブ ---- 97
(1) 相関 2 軸補間ドライブ ---- 97
(2) 任意軸補間ドライブ ---- 98
(3) 直線補間ドライブ ---- 100
(4) 円弧補間ドライブ ---- 101
(5) 線速一定制御 ---- 104
4-1-6. ドライブ CHANGE 機能 ---- 106
(1) UP/DOWN/CONST ドライブ CHANGE 機能 ---- 106
(2) SPEED CHANGE 機能 ---- 108
(3) RATE CHANGE 機能 ---- 109
(4) INDEX CHANGE 機能 ---- 110
4-1-7. 読み出し機能 ---- 113
(1) カウントデータのラッチデータ読み出し ---- 113
4-2. カウンタ仕様 ---- 114
4-2-1. リングカウンタ機能 ---- 114
4-2-2. カウントデータのラッチ・クリア機能 ---- 115
4-3. I/O 仕様 ---- 116
4-3-1. その他の I/O PORT ---- 116
(1) SIGNAL I/O 信号 ---- 116

５． 付録

5-1. 初期仕様一覧 ---- 117
(1) 応用設定 ---- 117
(2) 応用ドライブパラメータ ---- 118
5-2. 関数一覧 ---- 119
5-3. ドライブコマンド一覧 ---- 123
(1) 汎用コマンド ---- 123
(2) 特殊コマンド ---- 125

本版で改訂された主な箇所

1．概要

1-1. 特徴

● AL-Ⅱシリーズは、装置の分散化や補助軸の追加に柔軟且つ簡易に対応できるステッピングモータ、サーボモータ、デジタル I/O、アナログ I/O をコントロールする弊社オリジナルの高速シリアル通信システムです。

・AL-Ⅱシリーズは、20Mbps/50m または 10Mbps/100m の絶縁型高速シリアル通信です。
これにより、従来ボードコントローラに匹敵する性能(弊社比)でパソコンシステムの省配線化が図れます。

・Windows 用デバイスドライバ関数は、弊社製 PCI ボードコントローラ C-VX870 シリーズ(デバイス関数)、および USB シリーズ(デバイス関数とユニット関数)間で互いに移行が容易な仕様です。

● MPL-34-02v1.00/AL2W32、および MPL-35-02v1.00/AL2W64 は、Windows パソコン上で AL-Ⅱシリーズの PCI マスターまたは PCI Express マスターを介して、弊社製ステッピング＆ サーボモータコントローラ AL-Ⅱシリーズの製品を動作させるための DLL ベースのドライバ関数です。
当デバイスドライバは、MCC09 搭載した新たなスレーブコントローラ 2C-7760 に対応したバージョンアップ品ですが、MCC07 搭載製品仕様にも対応しています。

・Windows 32 ビット対応版が MPL-34-02v1.00/AL2W32 です。

・Windows 64 ビット対応版が MPL-35-02v1.00/AL2W64 です。
MPL-35-02v1.00/AL2W64 は、画像処理などの高速化を目的とした Windows 64 ビット環境のモーションおよび I/O システムを可能にします。

● 取扱説明書で解説している標準仕様の他に、この応用機能編では、主に以下を解説します。
・コマンド予約機能
・ドライブ CHANGE 機能
・外部トリガ出力機能

以下についての詳細は、デバイスドライバ取扱説明書をご覧ください。
・サポート OS や言語他
・ソフト開発に必要なファイル
・デバイスドライバの制限事項
・標準的な関数、コマンド、機能の解説
・全体の実行シーケンス等

2．関数リファレンス

当デバイスドライバ仕様は、MCC07 および MCC09 の搭載製品ならびにデジタル I/O やアナログ I/O を含む関数を網羅しています。

スレーブユニットに MCC07 および MCC09 搭載製品が混在したシステム構成も可能です。
原則、関数仕様は共通性がありますので、搭載チップ毎のスレーブユニットによって、ユーザアプリケーションを分ける必要はありません。

なお、MCC07 と MCC09 搭載の製品の差異を明らかにできるように、本書では MCC07 搭載製品を対象とした製品の仕様を示します。

2-1. MCC PORT アクセス関数の移行について

本書で説明するデバイス関数は、主に従来より対応している関数群です。
これらの関数は、下記に推奨している標準関数で代用することが可能です。
下記の従来関数から推奨関数に移行することで、特に、マルチスレッドプログラミングにおける、ユーザアプリケーション側での排他処理を不要とすることができます。

従来関数群
・DRIVE COMMAND データ構造体書き込み関数(MC07_IWDrive)
・DRIVE DATA データ構造体書き込み関数(MC07_IWData)
・DRIVE DATA 32 ビット書き込み関数(MC07_LWData)
・DRIVE DATA1 PORT 書き込み関数(MC07_BWDriveData1)
・DRIVE DATA2 PORT 書き込み関数(MC07_BWDriveData2)

⇩

推奨関数
・DRIVE COMMAND 32 ビット書き込み関数(MC07_LWDrive)

従来関数群
・DRIVE DATA データ構造体読み出し関数(MC07_IRDrive)
・DRIVE DATA 32 ビット読み出し関数(MC07_LRDrive)
・DRIVE DATA1 PORT 読み出し関数（MC07_BRDriveData1)
・DRIVE DATA2 PORT 読み出し関数（MC07_BRDriveData2)

⇩

推奨関数
・DRIVE COMMAND 32 ビット書き込み関数／読み出し関数(MC07_LWRDrive)

* MCC07 製品では、DRIVE COMMAND PORT 書き込み関数／読み出し関数(MC07_BWRDrive) は対応していません。

2-2. デバイス関数

2-2-1. MCC PORT アクセス関数

DRIVE DATA 32ビット書き込み関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機　能

指定されたデバイスのDRIVE DATA1 PORT、DRIVE DATA2 PORTにデータを書き込みます。

書　式

C言語　BOOL MCO7_LWData(DWORD *hDev*, DWORD **pData*, MCO7_S_RESULT **psResult*);

VB　Function MCO7_LWData(ByVal *hDev* As Long, ByRef *pData* As Long,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

VB.NET　Function MCO7_LWData(ByVal *hDev* As Integer, ByRef *pData* As Integer,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

C#.NET　bool MCO7.LWData(uint *hDev*, ref uint *pData*, ref MCO7_S_RESULT *psResult*);

引　数

hDev　・・・ デバイスハンドルを指定します。
pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
・変数の上位16ビットは、DRIVE DATA2 PORTに書き込まれます。
・変数の下位16ビットは、DRIVE DATA1 PORTに書き込まれます。
psResult　・・・ この関数を実行した結果を格納するためのRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

DRIVE DATA1 PORT書き込み関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機　能

指定されたデバイスのDRIVE DATA1 PORTにデータを書き込みます。

書　式

C言語　BOOL MCO7_BWDriveData1(DWORD *hDev*, WORD **pData*, MCO7_S_RESULT **psResult*);

VB　Function MCO7_BWDriveData1(ByVal *hDev* As Long, ByRef *pData* As Integer,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

VB.NET　Function MCO7_BWDriveData1(ByVal *hDev* As Integer, ByRef *pData* As Short,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

C#.NET　bool MCO7.BWDriveData1(uint *hDev*, ref ushort *pData*, ref MCO7_S_RESULT *psResult*);

引　数

hDev　・・・ デバイスハンドルを指定します。
pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
psResult　・・・ この関数を実行した結果を格納するためのRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

DRIVE DATA2 PORT書き込み関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機　能

指定されたデバイスのDRIVE DATA2 PORTにデータを書き込みます。

書　式

C言語　BOOL MCO7_BWDriveData2(DWORD *hDev*, WORD **pData*, MCO7_S_RESULT **psResult*);

```
VB      Function MCO7_BWDriveData2(ByVal hDev As Long, ByRef pData As Integer,
                ByRef psResult As MCO7_S_RESULT) As Boolean

VB.NET  Function MCO7_BWDriveData2(ByVal hDev As Integer, ByRef pData As Short,
                ByRef psResult As MCO7_S_RESULT) As Boolean

C#.NET  bool MCO7.BWDriveData2(uint hDev, ref ushort pData, ref MCO7_S_RESULT psResult);
```

引　数

hDev　　･･･ デバイスハンドルを指定します。
pData　　･･･ 書き込むデータが格納されている変数のポインタを指定します。
psResult　･･･ この関数を実行した結果を格納するためのRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

DRIVE DATA 32ビット読み出し関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機 能

指定されたデバイスのDRIVE DATA1 PORT、DRIVE DATA2 PORTの内容を読み出します。

書 式

C言語 BOOL MCO7_LRDrive(DWORD *hDev*, DWORD **pData*, MCO7_S_RESULT **psResult*);

VB Function MCO7_LRDrive(ByVal *hDev* As Long, ByRef *pData* As Long,
ByRef *psResult* As MCO7_S_RESULT)As Boolean

VB.NET Function MCO7_LRDrive(ByVal *hDev* As Integer, ByRef *pData* As Integer,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

C#.NET bool MCO7.LRDrive(uint *hDev*, ref uint *pData*, ref MCO7_S_RESULT *psResult*);

引 数

hDev ･･･ デバイスハンドルを指定します。
pData ･･･ 読み出した内容を格納するための変数のポインタを指定します。
・DRIVE DATA2 PORTの内容が変数の上位16ビットに格納されます。
・DRIVE DATA1 PORTの内容が変数の下位16ビットに格納されます。
psResult ･･･ この関数を実行した結果を格納するためのRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

DRIVE DATA1 PORT読み出し関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機　能

指定されたデバイスのDRIVE DATA1 PORTの内容を読み出します。

書　式

C言語　BOOL MCO7_BRDriveData1(DWORD *hDev*, WORD **pData*, MCO7_S_RESULT **psResult*);

VB　Function MCO7_BRDriveData1(ByVal *hDev* As Long, ByRef *pData* As Integer,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

VB.NET　Function MCO7_BRDriveData1(ByVal *hDev* As Integer, ByRef *pData* As Short,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

C#.NET　bool MCO7.BRDriveData1(uint *hDev*, ref ushort *pData*, ref MCO7_S_RESULT *psResult*);

引　数

hDev　・・・ デバイスハンドルを指定します。
pData　・・・ 読み出した内容を格納するための変数のポインタを指定します。
psResult　・・・ この関数を実行した結果を格納するためのRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

DRIVE DATA2 PORT読み出し関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機　能

指定されたデバイスのDRIVE DATA2 PORTの内容を読み出します。

書　式

C言語　BOOL MCO7_BRDriveData2(DWORD *hDev*, WORD **pData*, MCO7_S_RESULT **psResult*);

VB　Function MCO7_BRDriveData2(ByVal *hDev* As Long, ByRef *pData* As Integer,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

VB.NET　Function MCO7_BRDriveData2(ByVal *hDev* As Integer, ByRef *pData* As Short,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

C#.NET　bool MCO7.BRDriveData2(uint *hDev*, ref ushort *pData*, ref MCO7_S_RESULT *psResult*);

引　数

hDev　・・・ デバイスハンドルを指定します。
pData　・・・ 読み出した内容を格納するための変数のポインタを指定します。
psResult　・・・ この関数を実行した結果を格納するためのRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

データ構造体

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機 能

DRIVE DATA1 PORT、DRIVE DATA2 PORTに書き込むデータを格納します。

書 式

C言語
```
typedef struct _MC07_S_DATA{
      WORD MC07_Data[4];
} MC07_S_DATA
```

VB
```
Type MC07_S_DATA
    MC07_Data(1 To 4) As Integer
End Type
```

VB.NET
```
Structure MC07_S_DATA
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=4)> Public MC07_Data() As Short
    Public Sub Initialize()
        ReDim MC07_Data(3)
    End Sub
End Structure
```

C#.NET
```
struct MC07_S_DATA
{
    [MarshalAs( UnmanagedType.ByValArray, SizeConst=4 )] public ushort[] MC07_Data;
    MC07_S_DATA( ushort dummy )
    {
        MC07_Data = new ushort[4];
    }
}
```

メンバ

次に示すメンバは、C言語で表記しています。C言語のMC07_Data[0]～MC07_Data[3]は、Visual Basicでは
MC07_Data(1)～MC07_Data(4)、Visual Basic.NETではMC07_Data(0)～MC07_Data(3)、
C#.NETではMC07_Data[0]～MC07_Data[3]に対応します。

MC07_Data[0] ･･･ DRIVE DATA1 PORTの内容を格納します。
MC07_Data[1] ･･･ DRIVE DATA2 PORTの内容を格納します。
MC07_Data[2] ･･･ 将来の拡張用です。
MC07_Data[3] ･･･ 将来の拡張用です。

*VB.NETでは、配列を含む構造体は、Initializeメソッドにより配列を作成します。
*C#.NETでは、配列を含む構造体は、コンストラクタにより配列を作成します。
コンストラクタの引数dummyは何を指定しても無効です。

DRIVE COMMAND データ構造体書き込み関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機 能

指定されたデバイスのDRIVE DATA1 PORT、DRIVE DATA2 PORTにデータ構造体の内容を書き込んだ後、
DRIVE COMMAND PORTにコマンドコードを書き込みます。

書 式

C言語 BOOL MCO7_IWDrive(DWORD *hDev*, WORD *Cmd*, MCO7_S_DATA **psData*, MCO7_S_RESULT **psResult*);

VB Function MCO7_IWDrive(ByVal *hDev* As Long, ByVal *Cmd* As Integer, ByRef *psData* As MCO7_S_DATA,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

VB.NET Function MCO7_IWDrive(ByVal *hDev* As Integer, ByVal *Cmd* As Short, ByRef *psData* As MCO7_S_DATA,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

C#.NET bool MCO7.IWDrive(uint *hDev*, ushort *Cmd*, ref MCO7_S_DATA *psData*,
ref MCO7_S_RESULT *psResult*);

引 数

hDev ･･･ デバイスハンドルを指定します。
Cmd ･･･ 書き込むコマンドコードを指定します。
psData ･･･ 書き込むデータが格納されているデータ構造体のポインタを指定します。
psResult ･･･ この関数を実行した結果を格納するためのRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

DRIVE DATA データ構造体書き込み関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機 能

指定されたデバイスのDRIVE DATA1 PORT、DRIVE DATA2 PORTにデータを書き込みます。

書 式

C言語 BOOL MCO7_IWData(DWORD *hDev*, MCO7_S_DATA **psData*, MCO7_S_RESULT **psResult*);

VB Function MCO7_IWData(ByVal *hDev* As Long, ByRef *psData* As MCO7_S_DATA,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

VB.NET Function MCO7_IWData(ByVal *hDev* As Integer, ByRef *psData* As MCO7_S_DATA,
ByRef *psResult* As MCO7_S_RESULT) As Boolean

C#.NET bool MCO7.IWData(uint *hDev*, ref MCO7_S_DATA *psData*,
ref MCO7_S_RESULT *psResult*);

引 数

hDev ･･･ デバイスハンドルを指定します。
psData ･･･ 書き込むデータが格納されているデータ構造体のポインタを指定します。
psResult ･･･ この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

DRIVE DATA データ構造体読み出し関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機　能

指定されたデバイスのDRIVE DATA1 PORT、DRIVE DATA2 PORTの内容を読み出します。

書　式

C言語　BOOL MC07_IRDrive(DWORD *hDev*, MC07_S_DATA **psData*, MC07_S_RESULT **psResult*);

VB　Function MC07_IRDrive(ByVal *hDev* As Long, ByRef *psData* As MC07_S_DATA,
ByRef *psResult* As MC07_S_RESULT)As Boolean

VB.NET　Function MC07_IRDrive(ByVal *hDev* As Integer, ByRef *psData* As MC07_S_DATA,
ByRef *psResult* As MC07_S_RESULT) As Boolean

C#.NET　bool MC07.IRDrive(uint *hDev*, ref MC07_S_DATA *psData*, ref MC07_S_RESULT *psResult*);

引　数

hDev　・・・ デバイスハンドルを指定します。
psData　・・・ 読み出した内容を格納するためのデータ構造体のポインタを指定します。
psResult　・・・ この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

データセット関数

AL2A

―

機　能

32ビットデータを次の形式でデータ構造体に格納します。

引数psDataで示されるデータ構造体のメンバ*MCO7_Data[0]* (C言語表記)[各種DATA1 PORTに対応]

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					引数 Data の下位 16 ビット(D15 ～ D0)								→	D0

引数psDataで示されるデータ構造体のメンバ*MCO7_Data[0]* (C言語表記)[各種DATA2 PORTに対応]

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←					引数 Data の上位 16 ビット(D31 ～ D16)								→	D16

書　式

C言語　VOID MCO7_SetData(DWORD *Data*, MCO7_S_DATA **psData*);

VB　Sub MCO7_SetData(ByVal *Data* As Long, ByRef *psData* As MCO7_S_DATA)

VB.NET　Sub MCO7_SetData(ByVal *Data* As Integer, ByRef *psData* As MCO7_S_DATA)

C#.NET　void MCO7.SetData(uint *Data*, ref MCO7_S_DATA *psData*);

引　数

Data　・・・ 32ビットのデータを指定します。
psData　・・・ データ構造体のポインタを指定します。

戻り値

この関数に、戻り値はありません。

データゲット関数

AL2A

―

機　能

データ構造体の内容を、次の形式で32ビットデータに変換し返します。

変換後の32ビットデータ

D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
D15	←	引数psDataで示されるデータ構造体のメンバMCO7_Data[1](C言語表記) [各種DATA2 PORTに対応]												→	D0

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←	引数psDataで示されるデータ構造体のメンバMCO7_Data[0](C言語表記) [各種DATA1 PORTに対応]												→	D0

書　式

C言語　DWORD MCO7_GetData(MCO7_S_DATA *_psData_);

VB　Function MCO7_GetData(ByRef _psData_ As MCO7_S_DATA) As Long

VB.NET　Function MCO7_GetData(ByRef _psData_ As MCO7_S_DATA) As Integer

C#.NET　Uint MCO7.GetData(ref MCO7_S_DATA _psData_);

引　数

psData　　･･･ データ構造体のポインタを指定します。

戻り値

32ビットに変換されたデータを返します。

2-2-2. コマンド予約機能関数

MCC07 の汎用コマンドは、コマンド予約機能によりコマンドを予約レジスタに書き込むことができます。
コマンド予約機能で汎用コマンドを書き込むときは、DRIVE STATUS1 PORT 内の ERROR=0 および COMREG FL BIT=0 を確認してから汎用コマンドを書き込みます。
また、汎用コマンドをコマンド予約機能で書き込みした後に、予約コマンドの空き待ちを行います。
COMREG NOT FULL WAIT 関数は、この汎用コマンドを予約コマンドとして書き込みした後の終了待ちするときに用いる関数です。

COMREG NOT FULL WAIT関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機　能

指定されたデバイスがコマンドレジスタ予約可能な状態(DRIVE STATUS1 PORT COMREG FULL BIT = 0) になるまで待機します。
・最大待ち時間を超えるとエラー終了します。
・待機中にDRIVE STATUS1 PORTのERROR = 1 を検出したときもエラー終了します。

書　式

C言語　BOOL MC07_BWaitComregNotFull(DWORD *hDev*, WORD *WaitTime*, MC07_S_RESULT **psResult*);

```
VB      Function MC07_BWaitComregNotFull(ByVal hDev As Long, ByVal WaitTime As Integer,
                ByRef psResult As MC07_S_RESULT) As Boolean

VB.NET  Function MC07_BWaitComregNotFull(ByVal hDev As Integer, ByVal WaitTime As Short,
                ByRef psResult As MC07_S_RESULT) As Boolean
```

C#.NET　bool MC07.BWaitComregNotFull(uint *hDev*, ushort *WaitTime*, ref MC07_S_RESULT *psResult*);

引　数

hDev　　･･･ デバイスハンドルを指定します。
WaitTime　･･･ 最大待ち時間を1ms単位で設定します。
　　0を指定すると無限に待機します。
psResult　･･･ この関数を実行した結果を格納するためのRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

DRIVE COMMAND バッファー書き込み関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機 能

コマンドデータ構造体の配列に格納されているコマンド、データを指定されたデバイスのDRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE COMMAND PORTに指定された個数書き込みます。

書 式

C言語 BOOL MCO7_LWDriveBuf(DWORD *hDev*, MCO7_S_COMMAND_DATA *CmdDataBuf[]*, WORD *Count*,
MCO7_S_RESULT **psResult*);

VB Function MCO7_LWDriveBuf(ByVal *hDev* As Long, ByRef *CmdDataBuf* As MCO7_S_COMMAND_DATA,
ByVal *Count* As Integer, ByRef *psResult* As MCO7_S_RESULT) As Boolean

VB.NET Function MCO7_LWDriveBuf(ByVal *hDev* As Integer, ByVal *CmdDataBuf*() As MCO7_S_COMMAND_DATA,
ByVal *Count* As Short, ByRef *psResult* As MCO7_S_RESULT) As Boolean

C#.NET bool MCO7.LWDriveBuf(uint *hDev*, MCO7_S_COMMAND_DATA[] *CmdDataBuf*, ushort *Count*,
ref MCO7_S_RESULT *psResult*);

引 数

hDev ･･･ デバイスハンドルを指定します。
CmdDataBuf ･･･ コマンドデータ構造体の配列に指定します。
CmdDataBuf[0] ･･･ 1番目に書き込むコマンドコード構造体を格納します。
･･･
CmdDataBuf[7] ･･･ 8番目に書き込むコマンドコード構造体を格納します。
CmdDataBuf[8]～CmdDataBuf[15] ･･･ 将来の拡張用です。
Count ･･･ コマンドコード構造体の配列に格納されている要素の個数(最大8個)を指定します。
psResult ･･･ この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

2-2-3. 円弧補間の演算関数

円弧補間短軸PULSE数ゲット関数

AL2A

2C-771v1 | 2C-776Av1 | 2CD-7710v1 | 2CD-7713v1

機　能

指定された円弧の中心点、目的地、回転方向から、目的地までの短軸パルス数を計算します。

座標は、モータの現在位置を座標中心(0,0)とした相対アドレスです。
計算方法については、「4-1-5．補間ドライブ（4）円弧補間ドライブ」をご覧ください。
注．次の場合、関数がエラー終了します。
・円弧の中心点座標が(0，0)、または中心点と目的地が同一座標の場合
・円弧補間で求めた短軸PULSE数が-2,147,483,648～+2,147,483,647の範囲内でない場合
・円弧補間で指定出来ない目的地座標が指定された場合

書　式

C言語
```
BOOL MCO7_GetCirShortPulse( WORD Dir, MCO7_S_XY_POSITION *psCenterPosition,
        MCO7_S_XY_POSITION *psTargetPosition,
        LONG *pShortPulse, MCO7_S_RESULT *psResult );
```

VB
```
Function MCO7_GetCirShortPulse(ByVal Dir As Integer,
        ByRef psCenterPosition As MCO7_S_XY_POSITION,
        ByRef psTargetPosition As MCO7_S_XY_POSITION, ByRef pShortPulse As Long,
        ByRef psResult As MCO7_S_RESULT) As Boolean
```

VB.NET
```
Function MCO7_GetCirShortPulse(ByVal Dir As Short,
        ByRef psCenterPosition As MCO7_S_XY_POSITION,
        ByRef psTargetPosition As MCO7_S_XY_POSITION, ByRef pShortPulse As Integer,
        ByRef psResult As MCO7_S_RESULT) As Boolean
```

C#.NET
```
bool MCO7.GetCirShortPulse(ushort Dir, ref MCO7_S_XY_POSITION psCenterPosition,
        ref MCO7_S_XY_POSITION psTargetPosition, ref int pShortPulse,
        ref MCO7_S_RESULT psResult);
```

引　数

Dir　　･･･　回転方向を指定します。

指定できる値	意味
MCO7_CCW	-(CCW)方向
MCO7_CW	+(CW)方向

psCenterPosition　･･･ 中心点のX-Y相対座標が格納されているポジション構造体のポインタを指定します。
(指定できる値の範囲はX座標･Y座標ともに、-8,388,607～+8,388,607です。)
中心点のX-Y座標は、現在位置を座標の中心(0,0)とした相対座標です。

psTargetPosition　･･･ 目的地のX-Y相対座標が格納されているポジション構造体のポインタを指定します。
(指定できる値の範囲はX座標･Y座標ともに、-16,777,214～+16,777,214です。)
目的地のX-Y座標は、現在位置を座標の中心(0,0)とした相対座標です。

pShortPulse　　･･･ 目的地の短軸パルス数を格納するための変数のポインタを指定します。

psResult　　･･･ この関数を実行した結果を格納するためのRESULT構造体のポインタを指定します。

戻り値

この関数を実行した結果、正常終了したときはTRUE、エラーが発生したときはFALSEを返します。

2-2-4. その他の関数

16ビット符号なし変換関数

AL2A

－

機　能

指定された16ビット符号付きデータを16ビット符号なしデータに変換します。

書　式

C言語　WORD MC07_Unsigned16(SHORT *Data*);

VB　－

VB.NET　Function MC07_Unsigned16(ByVal *Data* As Short) As UShort

C#.NET　ushort MC07.Unsigned16(short *Data*);

引　数

Data　・・・ 16ビット符号付きデータを指定します。

戻り値

16ビット符号なしデータを返します。

16ビット符号付き変換関数

AL2A

－

機　能

指定された16ビット符号なしデータを16ビット符号つきデータに変換します。

書　式

C言語　SHORT MCO7_Signed16(WORD *Data*);

VB　　－

VB.NET　Function MCO7_Signed16(ByVal *Data* As UShort) As Short

C#.NET　short MCO7.Signed16(ushort *Data*);

引　数

Data　　･･･ 16ビット符号なしデータを指定します。

戻り値

16ビット符号付きデータを返します。

32ビット符号なし変換関数

AL2A

－

機　能

指定された32ビット符号付きデータを32ビット符号なしデータに変換します。

書　式

C言語　DWORD MC07_Unsigned32(LONG *Data*);

VB　－

VB.NET　Function MC07_Unsigned32(ByVal *Data* As Integer) As UInteger

C#.NET　uint MC07.Unsigned32(int *Data*);

引　数

Data　・・・ 32ビット符号付きデータを指定します。

戻り値

32ビット符号なしデータを返します。

32ビット符号付き変換関数

AL2A

－

機　能

指定された32ビット符号なしデータを32ビット符号つきデータに変換します。

書　式

C言語　LONG MC07_Signed32(DWORD *Data*);

VB　－

VB.NET　Function MC07_Signed32(ByVal *Data* As UInteger) As Integer

C#.NET　int MC07.Signed32(uint *Data*);

引　数

Data　･･･ 32ビット符号なしデータを指定します。

戻り値

32ビット符号付きデータを返します。

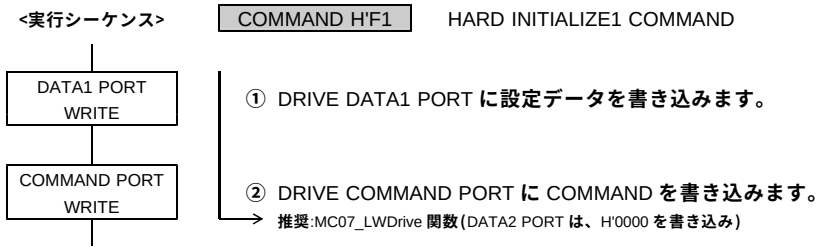
3．コマンド仕様

3-1. ドライブコマンド

3-1-1. 入出力仕様の設定

(1) HARD INITIALIZE1

ステータス信号(SOUT 信号)に出力する機能を設定します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。


<実行シーケンス>
COMMAND H'F1
HARD INITIALIZE1 COMMAND
DATA1 PORT WRITE
COMMAND PORT WRITE
① DRIVE DATA1 PORT に設定データを書き込みます。
② DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
1	1	1	0	1	1	1	0

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	0	SOUT TYPE3	SOUT TYPE2	SOUT TYPE1	SOUT TYPE0

●電源投入後の初期値は H'EE21（アンダーライン側）です。

D3--D0：SOUT TYPE3--0　　初期値 ＝ CNTINT　出力

SOUT 信号に出力するステータスを選択します。

TYPE3	TYPE2	TYPE1	TYPE0	SOUT 信号に出力する機能	
0	0	0	0	ADRINT	カウンタ割り込み要求の ADRINT
0	0	0	1	CNTINT	カウンタ割り込み要求の CNTINT
0	0	1	0	DFLINT	カウンタ割り込み要求の DFLINT
0	0	1	1	RDYINT	コマンド終了割り込み要求の RDYINT
0	1	0	0	STBY	STATUS1 の STBY フラグ
0	1	0	1	nDRIVE	STATUS1 の DRIVE フラグの反転
0	1	1	0	nSPEED CBUSY	STATUS5 の SPEED CBUSY フラグの反転
0	1	1	1	nINDEX CBUSY	STATUS5 の INDEX CBUSY フラグの反転
1	0	0	0	UP	STATUS1 の UP フラグ
1	0	0	1	DOWN	STATUS1 の DOWN フラグ
1	0	1	0	CONST	STATUS1 の CONST フラグ
1	0	1	1	EXT PULSE	STATUS1 の EXT PULSE フラグ
1	1	0	0	nPULSE MASK	STATUS2 の PULSE MASK フラグの反転
1	1	0	1	ORG SIGNAL	STATUS2 の ORG SIGNAL フラグ
1	1	1	0	汎用出力	汎用出力として使用する
1	1	1	1	PULSE OVF	STATUS4 の PULSE OVF フラグ

「汎用出力」を選択した場合は、SIGNAL OUT コマンドで出力レベルを操作します。

2C-776Av1 は、ステータス信号(SOUT 信号)を SIGNAL I/O コネクタから外部出力できます。
外部出力可能な軸は下記の 2 軸です。

外部出力信号	出力可能な軸
SOUT0 信号	X 軸の SOUT
SOUT1 信号	Y 軸の SOUT

*2C-776Av1 以外の製品は、ステータス出力はできません。

(2) HARD INITIALIZE4

CWLM, CCWLM 信号、$\overline{\text{DEND}}$/$\overline{\text{PO}}$ 信号 および DALM 信号入力のデジタルフィルタ機能の時定数を設定します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。

<実行シーケンス>

COMMAND H'F4 HARD INITIALIZE4 COMMAND

DATA1 PORT WRITE

① DRIVE DATA1 PORT に設定データを書き込みます。

COMMAND PORT WRITE

② DRIVE COMMAND PORT に COMMAND を書き込みます。

推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	SERVO FILTER2	SERVO FILTER1	SERVO FILTER0	–	0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
–	LIMIT FILTER2	LIMIT FILTER1	LIMIT FILTER0	–	0	0	0

●電源投入後の初期値は H'0000 (アンダーライン側) です。

D6--D4 ：LIMIT FILTER2--0 ：CWLM, CCWLM 信号入力の選択
D14--D12 ：SERVO FILTER2--0 ：$\overline{\text{DEND}}$/$\overline{\text{PO}}$, DALM 信号入力の選択
D2--D0 ：0 に設定する
D10--D8 ：0 に設定する

デジタルフィルタ機能の時定数を選択します。

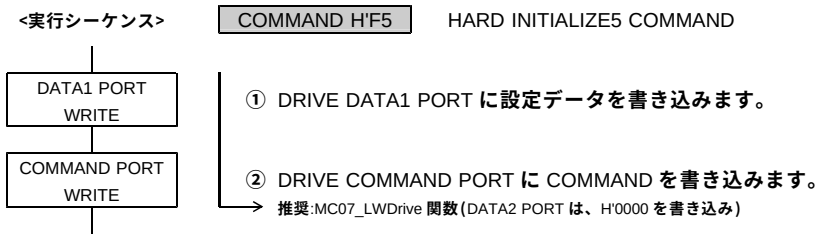
FLT2	FLT1	FLT0	デジタルフィルタ
0	0	0	0 ～ 50 ns
0	0	1	50 µs
0	1	0	100 µs
0	1	1	200 µs
1	0	0	500 µs
1	0	1	1.0 ms
1	1	0	5.0 ms
1	1	1	10.0 ms

(誤差：+10, -0 µs)

●コントローラの入力回路には、ハード的なノイズフィルタが入っており、原則設定不要です。
環境下によって不要な信号を拾うようなとき、デジタルフィルタを設定してください。
但し、デジタルフィルタの設定により、信号の応答性は低下します。

(3) HARD INITIALIZE5

ORG 信号、NORG 信号, ± ZORG 信号入力のデジタルフィルタ機能の時定数を設定します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。


<実行シーケンス>
COMMAND H'F5
HARD INITIALIZE5 COMMAND
DATA1 PORT WRITE
COMMAND PORT WRITE
① DRIVE DATA1 PORT に設定データを書き込みます。
② DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	0	0	0	–	0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
–	ZPO FILTER2	ZPO FILTER1	ZPO FILTER0	–	ORIGIN FILTER2	ORIGIN FILTER1	ORIGIN FILTER0

●電源投入後の初期値は H'0000 (アンダーライン側) です。

D2--D0 ：ORIGIN FILTER2--0 ：ORG,NORG 信号
D10--D8 ：設定禁止(0)
D14--D12 ：設定禁止(0)

デジタルフィルタ機能の時定数を選択します。

FLT2	FLT1	FLT0	デジタルフィルタ
0	0	0	0 ～ 50 ns
0	0	1	50 µs
0	1	0	100 µs
0	1	1	200 µs
1	0	0	500 µs
1	0	1	1.0 ms
1	1	0	5.0 ms
1	1	1	10.0 ms

(誤差：+10, -0 µs)

D6--D4 ：ZPO FILTER2--0 ：± ZORG 信号

デジタルフィルタ機能の時定数を選択します。

FLT2	FLT1	FLT0	デジタルフィルタ
0	0	0	0 ～ 50 ns
0	0	1	5 µs
0	1	0	10 µs
0	1	1	20 µs
1	0	0	50 µs
1	0	1	100 µs
1	1	0	500 µs
1	1	1	1.0 ms

(誤差：+1, -0 µs)

●コントローラの入力回路には、ハード的なノイズフィルタが入っており、原則設定不要です。
環境下によって不要な信号を拾うようなとき、デジタルフィルタを設定してください。
但し、デジタルフィルタの設定により、信号の応答性は低下します。

(4) HARD INITIALIZE6

± EA,± EB 信号入力のデジタルフィルタ機能の時定数を設定します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。

<実行シーケンス>

COMMAND H'F6　HARD INITIALIZE6 COMMAND

DATA1 PORT WRITE

① DRIVE DATA1 PORT に設定データを書き込みます。
設定範囲は、0 ～ 255（H'00 ～ H'FF）です。

COMMAND PORT WRITE

② DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	D7	← デジタルフィルタのデータ →						D0

●電源投入後の初期値は H'00（0 ～ 50 ns）です。

± EA, ± EB 信号入力のデジタルフィルタ機能の時定数を設定します。
デジタルフィルタの時定数 ＝ 設定データ x 50 ns（0 ～ 12.75 µs）です。
(誤差：+50, -0 ns)

●環境下によって不要な信号を拾うようなとき、デジタルフィルタを設定してください。
但し、デジタルフィルタの設定により、信号の応答性は低下します。

(5) HARD INITIALIZE7

CWLM, CCWLM 信号、FSSTOP 信号、DALM 信号の入力アクティブ論理を設定します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。

> ⚠ 注意
> 設定を誤ると停止機能が働かなくなります。
> 入力信号のアクティブ論理を設定するデータ部には、重要な停止信号などが割り付いています。指定された論理は間違えないようにしてください。

<実行シーケンス>

COMMAND H'F7　HARD INITIALIZE7 COMMAND

DATA1 PORT WRITE

① DRIVE DATA1 PORT に設定データを書き込みます。

COMMAND PORT WRITE

② DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
PAUSE ACTIVE(1)	1	ORG ACTIVE(1)	1	1	NORG ACTIVE(1)	1	FSSTOP ACTIVE(0)

D7	D6	D5	D4	D3	D2	D1	D0
1	DALM ACTIVE(0)	1	1	CCWLM ACTIVE(0)	CWLM ACTIVE(0)	FSSTOP ACTIVE(0)	1

●電源投入後の初期値は H'FEB1(アンダーライン側) です。
上記の値で、入力アクティブ論理は以下の初期値になっています。

D1,D8 ：FSSTOP ACTIVE
FSSTOP 入力信号のアクティブ論理を選択します。
0 ：正論理(ハイアクティブ)
1 ：負論理(ローアクティブ)
*FSSTOP 信号の論理を切り替えるときは、D1 と D8 ビットを必ず同じ値に設定してください。

D2 ：CWLM ACTIVE
CWLM 入力信号のアクティブ論理を選択します。
0 ：正論理(ハイアクティブ)
1 ：負論理(ローアクティブ)

D3 ：CCWLM ACTIVE
CCWLM 入力信号のアクティブ論理を選択します。
0 ：正論理(ハイアクティブ)
1 ：負論理(ローアクティブ)

D6 ：DALM ACTIVE
DALM 入力信号のアクティブ論理を選択します。
0 ：正論理(ハイアクティブ)
1 ：負論理(ローアクティブ)

上記の信号は、初期値 B 接点入力になっています。

D10 ：NORG ACTIVE
NORG 入力信号のアクティブ論理を選択します。
0 ：正論理(ハイアクティブ)
1 ：負論理(ローアクティブ)

D13 ：ORG ACTIVE
ORG 入力信号のアクティブ論理を選択します。
0 ：正論理(ハイアクティブ)
1 ：負論理(ローアクティブ)

D15 ：PAUSE ACTIVE
PAUSE 信号のアクティブ論理を選択します。
0 ：PAUSE=1 にして STBY にする
1 ：PAUSE=0 にして STBY を解除する

・D15 の PAUSE 信号のアクティブ論理を操作することで、コマンド予約機能による連続ドライブの設定と実行ができます。

・HARD INITIALIZE7 コマンドの実行で、各信号のアクティブ論理を変更します。
アクティブ論理を変更すると、変更した信号のデジタルフィルタ機能が動作します。
デジタルフィルタ機能の時定数経過後に、アクティブ論理の変更が確定します。

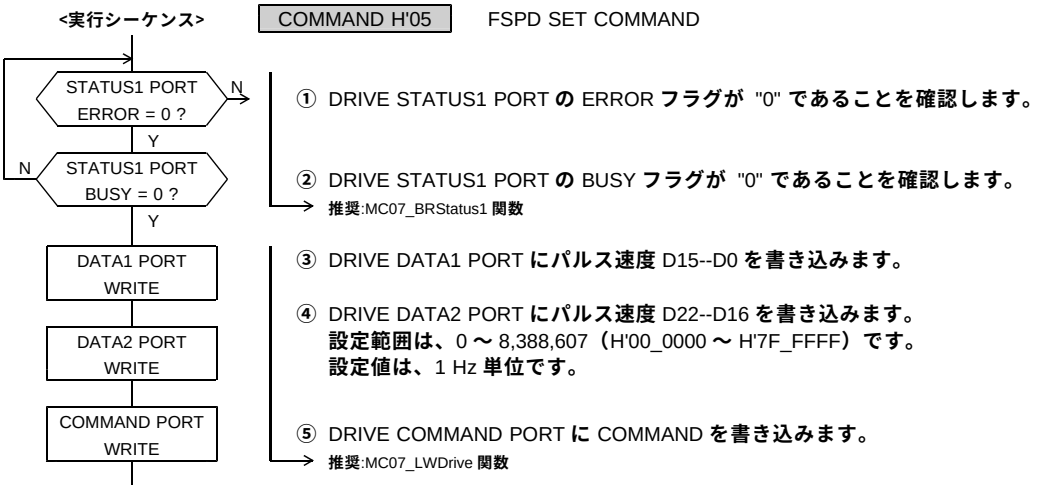
3-1-2. ドライブ パラメータの設定

ドライブのパラメータを設定します。各設定は変更が必要な場合に設定します。

※ SPEED・RATE 関数はこれらのコマンドを使用してドライブパラメータを設定しています。

(1) FSPD SET

ドライブパルス出力の第１パルス目のパルス周期（パルス速度）を設定します。


<実行シーケンス>
COMMAND H'05
FSPD SET COMMAND
STATUS1 PORT ERROR = 0 ?
N
Y
STATUS1 PORT BUSY = 0 ?
N
Y
DATA1 PORT WRITE
DATA2 PORT WRITE
COMMAND PORT WRITE
① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
③ DRIVE DATA1 PORT にパルス速度 D15--D0 を書き込みます。
④ DRIVE DATA2 PORT にパルス速度 D22--D16 を書き込みます。
設定範囲は、0 ～ 8,388,607（H'00_0000 ～ H'7F_FFFF）です。
設定値は、1 Hz 単位です。
⑤ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15 ←							FSPD							→	D0

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	D22 ←		FSPD			→	D16

●電源投入後の初期値は H'00_1388 （5,000 Hz ：１周期 200 μs） です。

*FSPD は、設定された値の半周期を経過してから第 1 パルス目のパルス出力を開始します。
初期値 200 μ s は、この周期が実際に第 1 パルス目の出力(立ち上がりエッジ)までの遅延時間になります。
よって、起動指令後に速くパルス出力を開始したいときは、ドライバ側の入力応答周波数に注意して、この FSPD の値を設定してください。

FSPD の設定値が "0" の場合は、"1" に補正します。

・第１パルスのパルス周期の計算式

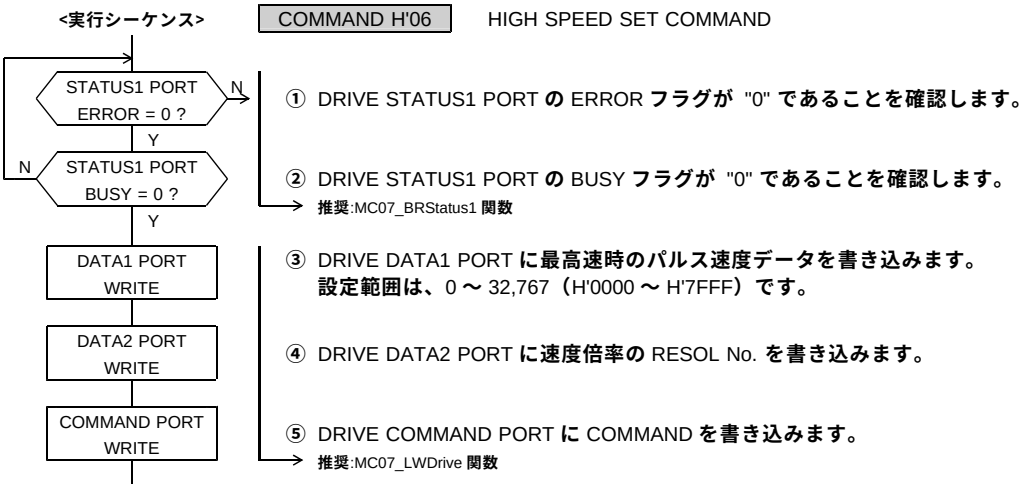
10,000,000 ／ FSPD ＝ 商 A ＋余り B → OFF 周期 ＝商 A x 50 ns
(10,000,000 ＋余り B)／ FSPD ＝ 商 C ＋余り D → ON 周期 ＝商 C x 50 ns

FSPD の設定値と実際に出力する第１パルスのパルス周期

FSPD の設定値	第１パルスのパルス周期（パルス速度）
8,388,607 ～ 6,666,667 Hz →	OFF 周期 = 50 ns ON 周期 = 50 ns （10,000,000 Hz）
6,666,666 ～ 5,000,001 Hz →	OFF 周期 = 50 ns ON 周期 = 100 ns（6,666,666 Hz）
5,000,000 ～ 4,000,001 Hz →	OFF 周期 = 100 ns ON 周期 = 100 ns（5,000,000 Hz）
4,000,000 ～ 3,333,334 Hz →	OFF 周期 = 100 ns ON 周期 = 150 ns（4,000,000 Hz）
3,333,333 ～ 2,857,143 Hz →	OFF 周期 = 150 ns ON 周期 = 150 ns（3,333,333 Hz）

(2) HIGH SPEED SET

加減速ドライブの最高速時のパルス速度データ(HSPD)を設定します。
加減速ドライブの速度データの速度倍率(RESOL)を設定します。


<実行シーケンス>
COMMAND H'06
HIGH SPEED SET COMMAND
STATUS1 PORT ERROR = 0 ?
N
Y
STATUS1 PORT BUSY = 0 ?
N
Y
DATA1 PORT WRITE
DATA2 PORT WRITE
COMMAND PORT WRITE
① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
③ DRIVE DATA1 PORT に最高速時のパルス速度データを書き込みます。
設定範囲は、0 ～ 32,767（H'0000 ～ H'7FFF）です。
④ DRIVE DATA2 PORT に速度倍率の RESOL No. を書き込みます。
⑤ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	D14 ←						HSPD							→	D0

●電源投入後の初期値は H'0BB8（3,000：3,000 Hz） です。

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	–	–	–	D3	RESOL No.		D0

●電源投入後の初期値は H'3（速度倍率＝ 1） です。

HSPD の設定値が "0" の場合は、HSPD を HSPD = LSPD に補正します。
・最高速時の速度（Hz）＝ HSPD x RESOL

減速ドライブと一定速ドライブの 1 パルス目は、FSPD の第 1 パルスです。
2 パルス目から HSPD x RESOL の速度になります。

RESOL No.を選択して、速度倍率（RESOL）を設定します。

RESOL No.	速度倍率 (RESOL)
H'0	0.1
H'1	0.2
H'2	0.5
H'3	1

RESOL No.	速度倍率 (RESOL)
H'4	2
H'5	5
H'6	10
H'7	20

RESOL No.	速度倍率 (RESOL)
H'8	50
H'9	100
H'A	200
H'B	200

RESOL No.	速度倍率 (RESOL)
H'C	200
H'D	200
H'E	200
H'F	200

・速度設定値 ＝ 速度データ × 速度倍率（RESOL）：0.1 ～ 6,553,400 Hz

(3) LOW SPEED SET

加減速ドライブの加速開始時のパルス速度データ(LSPD)を設定します。
加減速ドライブの減速終了時のパルス速度データ(ELSPD)を設定します。

<実行シーケンス>
COMMAND H'07
LOW SPEED SET COMMAND
STATUS1 PORT ERROR = 0 ?
N
Y
STATUS1 PORT BUSY = 0 ?
N
Y
DATA1 PORT WRITE
DATA2 PORT WRITE
COMMAND PORT WRITE
① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
③ DRIVE DATA1 PORT に加速開始時のパルス速度データを書き込みます。
設定範囲は、1 ～ 32,767（H'0001 ～ H'7FFF）です。
④ DRIVE DATA2 PORT に減速終了時のパルス速度データを書き込みます。
設定範囲は、0 ～ 32,767（H'0000 ～ H'7FFF）です。
⑤ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
−	D14 ←						LSPD							→	D0

●電源投入後の初期値は H'012C（300：300 Hz） です。

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
−	D14 ←						ELSPD							→	D0

●電源投入後の初期値は H'0000（LSPD と同じ） です。

LSPD の設定値が "0" の場合は、"1" に補正します。
・加速開始時の速度（Hz）＝ LSPD x RESOL

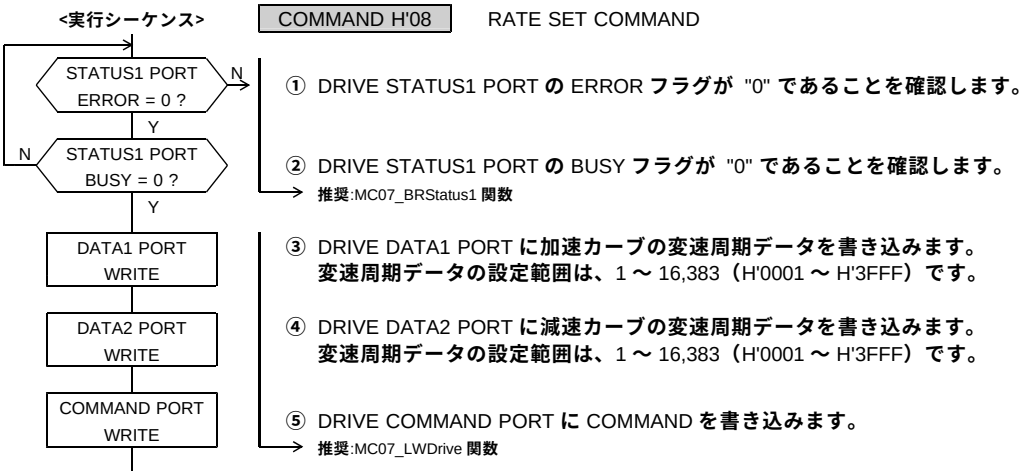
ELSPD の設定値が "0" の場合は、ELSPD を ELSPD = LSPD に補正します。
・減速終了時の速度（Hz）＝ ELSPD x RESOL

加減速ドライブと加速ドライブの 1 パルス目は、FSPD の第 1 パルスです。
2 パルス目から LSPD x RESOL の速度になります。

減速ドライブと一定速ドライブの１パルス目は、FSPD の第１パルスです。
2 パルス目から HSPD x RESOL の速度になります。

(4) RATE SET

加速カーブの変速周期データ(UCYCLE)を設定します。
減速カーブの変速周期データ(DCYCLE)を設定します。


<実行シーケンス>
COMMAND H'08
RATE SET COMMAND
STATUS1 PORT ERROR = 0 ?
N
Y
STATUS1 PORT BUSY = 0 ?
N
Y
DATA1 PORT WRITE
DATA2 PORT WRITE
COMMAND PORT WRITE
① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
③ DRIVE DATA1 PORT に加速カーブの変速周期データを書き込みます。
変速周期データの設定範囲は、1 ～ 16,383（H'0001 ～ H'3FFF）です。
④ DRIVE DATA2 PORT に減速カーブの変速周期データを書き込みます。
変速周期データの設定範囲は、1 ～ 16,383（H'0001 ～ H'3FFF）です。
⑤ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	D13 ←					UCYCLE								→ D0

●電源投入後の初期値は H'00C8（200：100 μs 周期） です。

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	D13 ←					DCYCLE								→ D0

●電源投入後の初期値は H'00C8（200：100 μs 周期） です。

UCYCLE の設定値が "0" の場合は、"1" に補正します。
・変速周期（μs）＝ UCYCLE x 0.5 μs ：0.5 μs ～ 8.1915 ms

DCYCLE の設定値が "0" の場合は、"1" に補正します。
・変速周期（μs）＝ DCYCLE x 0.5 μs ：0.5 μs ～ 8.1915 ms

(5) SCAREA SET

加速カーブの S 字変速領域データ(SUAREA)を設定します。
減速カーブの S 字変速領域データ(SDAREA)を設定します。

<実行シーケンス>
COMMAND H'09
SCAREA SET COMMAND
STATUS1 PORT ERROR = 0 ?
N
Y
STATUS1 PORT BUSY = 0 ?
N
Y
DATA1 PORT WRITE
DATA2 PORT WRITE
COMMAND PORT WRITE
① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
③ DRIVE DATA1 PORT に加速カーブの S 字変速領域データを書き込みます。
変速領域データの設定範囲は、0 ～ 16,383（H'0000 ～ H'3FFF）です。
④ DRIVE DATA2 PORT に減速カーブの S 字変速領域データを書き込みます。
変速領域データの設定範囲は、0 ～ 16,383（H'0000 ～ H'3FFF）です。
⑤ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
−	−	D13 ←					SUAREA								→ D0

●電源投入後の初期値は H'0000（0：SUAREA の変速領域なし） です。

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
−	−	D13 ←					SDAREA								→ D0

●電源投入後の初期値は H'0000 （0：SDAREA の変速領域なし） です。

SUAREA の設定値が "0" の場合は、
UCYCLE と RESOL による直線加速カーブのみで加速します。

SUAREA の設定値が "(HSPD − LSPD)／2" の場合は、S 字加速カーブのみで加速します。
SUAREA の設定値が "(HSPD − LSPD)／2" より大きい場合は、重複した変速領域を重ねます。
・SUAREA の変速領域（Hz） ＝ SUAREA × RESOL ：0 ～ (HSPD − LSPD) x RESOL ／ 2 Hz
・S 字加速開始部の変速領域（Hz） ＝ SUAREA × RESOL
・S 字加速終了部の変速領域（Hz） ＝ SUAREA × RESOL

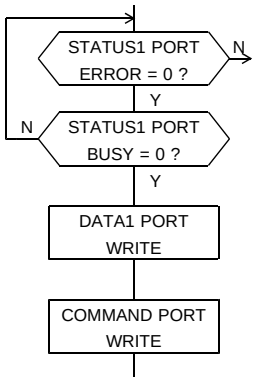
SDAREA の設定値が "0" の場合は、
DCYCLE と RESOL による直線減速カーブのみで減速します。

SDAREA の設定値が "(HSPD − ELSPD)／2" の場合は、S 字減速カーブのみで減速します。
SDAREA の設定値が "(HSPD − ELSPD)／2" より大きい場合は、以下のようになります。
減速停止指令の減速停止時には、重複した変速領域を重ねます。
INDEX ドライブの自動減速停止時には、SDAREA ＝(HSPD − ELSPD)／2 に補正します。
・SDAREA の変速領域（Hz） ＝ SDAREA × RESOL ：0 ～ (HSPD − ELSPD) x RESOL ／ 2 Hz
・S 字減速開始部の変速領域（Hz） ＝ SDAREA × RESOL
・S 字減速終了部の変速領域（Hz） ＝ SDAREA × RESOL

(6) DOWN PULSE ADJUST

INDEX ドライブの停止位置への減速停止動作時に有効です。
MCC07 が自動減速停止動作を開始する減速パルス数のオフセットパルス数を設定します。

<実行シーケンス>



① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。

② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数

③ DRIVE DATA1 PORT にパルス数 D15--D0 を書き込みます。
設定範囲は、-32,768 ～+32,767（H'8000 ～ H'7FFF）です。
負数の場合は 2 の補数表現にします。

④ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15 ←							オフセットパルス数								→ D0

●電源投入後の初期値は H'0001（1 パルス） です。

設定したオフセットパルス数は、MCC が自動検出する減速パルス数に加算します。
・オフセットパルス数を正数にすると、減速パルス数は増加します。
・オフセットパルス数を負数にすると、減速パルス数は減少します。

通常の INDEX ドライブでは、自動減速停止動作開始後に、停止位置を検出した時点で停止します。
・INDEX CHANGE 指令を検出した場合は、自動減速停止動作開始後に、終了速度に達してから停止位置を検出して停止します。
このため、負数のオフセットパルス数を設定している場合に INDEX CHANGE 指令を実行すると、停止位置を通過してから停止する可能性があります。
この場合は、ERROR STATUS の INDEX CHANGE ERROR = 1 になります。

3-1-3. ORIGIN ドライブの設定と実行

ORIGIN ドライブの動作仕様を設定して、ORIGIN ドライブを実行します。

※ ORIGIN 関数は、これらのコマンドを使用して、従来製品と同様の機械原点検出ドライブを実現しています。

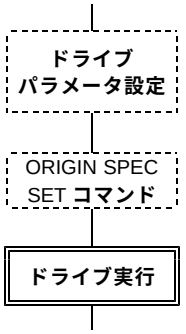
■ ORIGIN ドライブの検出工程

ドライブ工程は、ORIGIN SCAN ドライブと ORIGIN CONSTANT SCAN ドライブが選択できます。

● ORIGIN SCAN ドライブ
加減速ドライブのパラメータで、SCAN ドライブを行います。
ORG 検出信号の指定エッジを指定のカウント数検出すると減速停止します。

● ORIGIN CONSTANT SCAN ドライブ
JOG ドライブのパルス速度（JSPD）で、一定速ドライブを行います。
ORG 検出信号の指定エッジを指定のカウント数検出すると即時停止します。

■ ORIGIN ドライブの実行シーケンス



① ORIGIN ドライブに必要なパラメータを設定します。
・第 1 パルス出力周期
・加減速パラメータ または JOG パラメータ(JOG パルス速度)

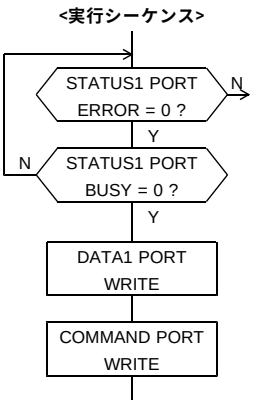
② ORIGIN ドライブの動作仕様を設定します。

②ドライブを実行します。

初期値および設定値に対して変更が必要な場合に設定します。

(1) ORIGIN SPEC SET

ORIGIN ドライブの動作仕様を設定します。



COMMAND H'0F　ORIGIN SPEC SET COMMAND

① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。

② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数

③ DRIVE DATA1 PORT に ORIGIN ドライブの動作仕様を書き込みます。

④ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	AUTO DRST ENABLE	ORG COUNT D3	ORG COUNT D2	ORG COUNT D1	ORG COUNT D0

D7	D6	D5	D4	D3	D2	D1	D0
–	–	ORIGIN START DIRECTION	ORG DETECT EDGE	ORG SIGNAL TYPE3	ORG SIGNAL TYPE2	ORG SIGNAL TYPE1	ORG SIGNAL TYPE0

●電源投入後の初期値は H'0003（アンダーライン側）です。

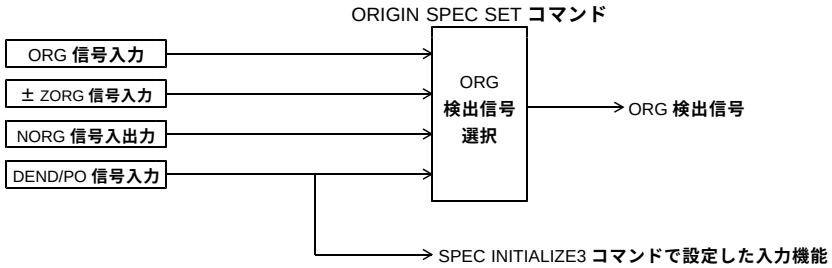
D3--D0：ORG SIGNAL TYPE3-0
ORIGIN ドライブで検出する ORG 検出信号を選択します。

TYPE3	TYPE2	TYPE1	TYPE0	ORG 検出信号
0	0	0	0	$\overline{\text{ORG}}$ 信号
0	0	0	1	± ZORG 信号
0	0	1	0	$\overline{\text{ORG}}$ 信号と ± ZORG 信号の AND（論理積）
0	0	1	1	$\overline{\text{ORG}}$ 信号と ± ZORG 信号の OR（論理和）
0	1	0	0	$\overline{\text{ORG}}$ 信号
0	1	0	1	$\overline{\text{DEND/PO}}$ 信号
0	1	1	0	$\overline{\text{ORG}}$ 信号と $\overline{\text{DEND/PO}}$ 信号の AND（論理積）
0	1	1	1	$\overline{\text{ORG}}$ 信号と $\overline{\text{DEND/PO}}$ 信号の OR（論理和）
1	0	0	0	$\overline{\text{NORG}}$ 信号
上記以外				設定禁止

・各信号入力のアクティブレベルを合成したものが、ORG 検出信号になります。

■ ORG 検出信号の構成

$\overline{\text{DEND/PO}}$ 信号を ORG 検出信号に選択した場合は、$\overline{\text{DEND/PO}}$ 信号の入力機能と ORG 検出信号の停止機能の両方が有効になります。



D4 ：ORG DETECT EDGE
ORG 検出信号の検出エッジを選択します。
0 ：ORG 検出信号の 0→1（アクティブ）エッジを検出する
1 ：ORG 検出信号の 1→0（OFF）エッジを検出する

D5 ：ORIGIN START DIRECTION
ORIGIN ドライブの起動方向を選択します。
0 ：－(CCW) 方向に起動する
1 ：＋(CW) 方向に起動する

D11-D8：ORG COUNT D3-D0

ORG 検出信号の検出エッジのカウント数を設定するビットです。

ORG 検出信号を指定のカウント数検出すると、ORIGIN ドライブの停止機能が動作します。

ORG COUNT D3--D0 = H'0　：１カウント目のエッジ検出で、停止機能を動作させる
ORG COUNT D3--D0 = H'1　：２カウント目のエッジ検出で、停止機能を動作させる
ORG COUNT D3--D0 = H'2　：３カウント目のエッジ検出で、停止機能を動作させる
：　：
ORG COUNT D3--D0 = H'F　：16 カウント目のエッジ検出で、停止機能を動作させる

D12 ：AUTO DRST ENABLE

SERVO SPEC SET コマンドで、DRST 信号を<サーボ対応>に設定している場合に有効です。

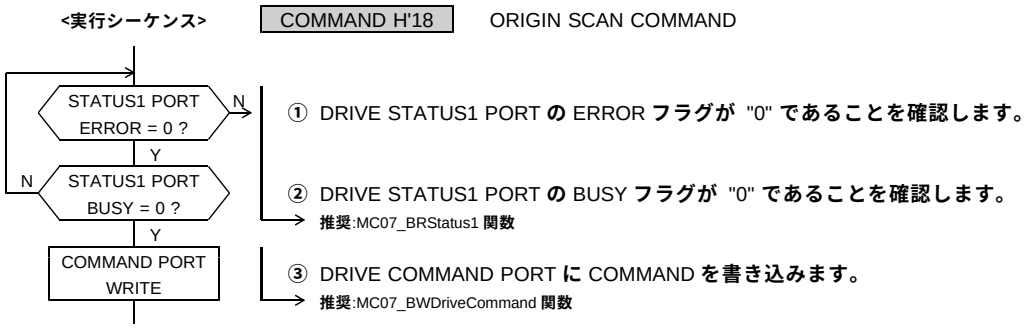
ORG 検出信号の停止機能が動作して ORIGIN ドライブを停止した時に、

DRST 信号を「出力する／出力しない」を選択します。

0 ：DRST 信号を出力しない
1 ：DRST 信号を出力する（10 ms 間アクティブレベルにする）

(2) ORIGIN SCAN

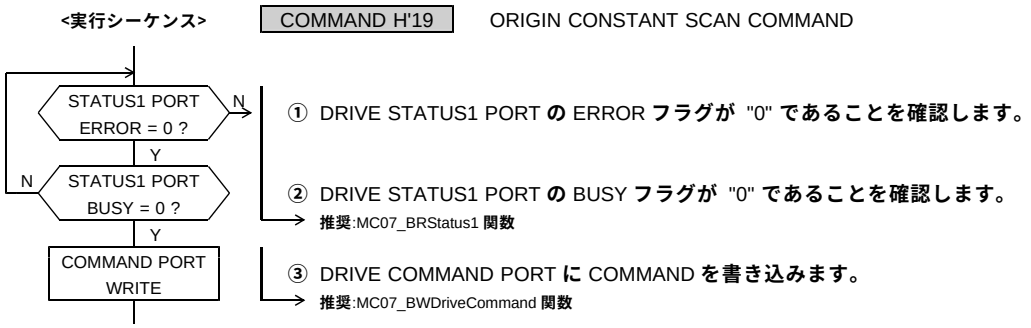
ORIGIN SCAN ドライブを実行します。



● ORIGIN SCAN ドライブ
加減速ドライブのパラメータで、SCAN ドライブを行います。
ORG 検出信号の指定エッジを指定のカウント数検出すると減速停止します。
AUTO DRST ENABLE = 1 の場合は、減速停止後に DRST 信号を出力します。

(3) ORIGIN CONSTANT SCAN

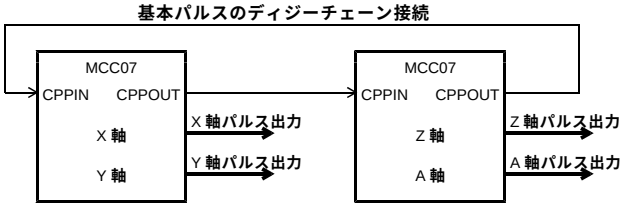
ORIGIN CONSTANT SCAN ドライブを実行します。



● ORIGIN CONSTANT SCAN ドライブ
JOG パルス速度（JSPD）で、一定速ドライブを行います。
ORG 検出信号の指定エッジを指定のカウント数検出すると即時停止します。
AUTO DRST ENABLE = 1 の場合は、即時停止後に DRST 信号を出力します。

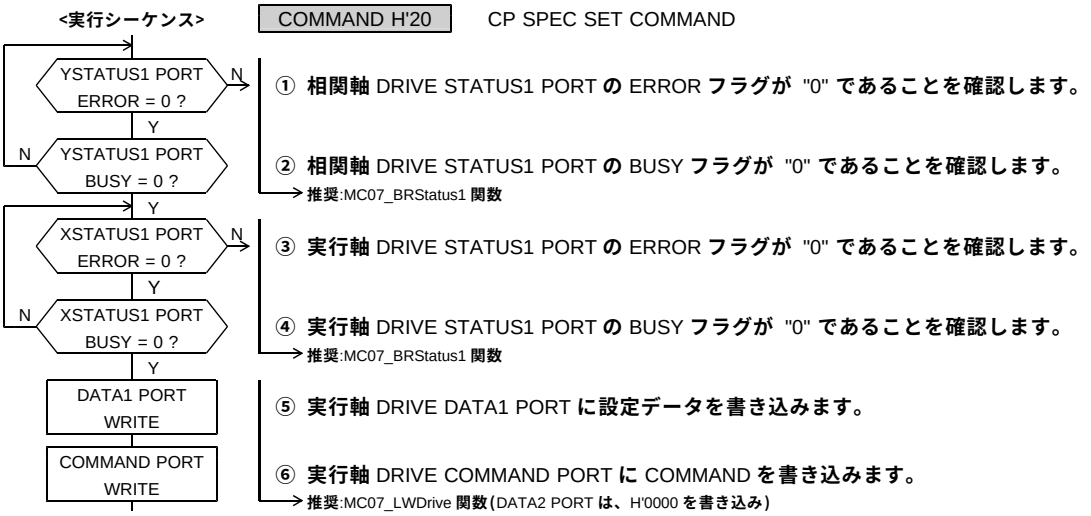
3-1-4. 任意軸補間ドライブの設定

MCC07 の CPPOUT 端子から出力するパルスを設定することで任意軸の補間を行うことができます。
4 軸コントローラ製品は、各軸 MCC07 の CPPOUT 端子と CPPIN 端子はディジーチェーン接続で繋がっています。
任意軸補間ドライブではメイン軸のチップが CPPOUT 端子に補間ドライブの基本パルスを出力します。
サブ軸のチップは基本パルスを CPPIN 端子から入力して CPPOUT 端子に出力します。



(1) CP SPEC SET

2 軸相関コマンドです。相関軸両軸が BUSY = 0, ERROR = 0 のときにコマンドを実行します。
相関 2 軸 1 チップの CPPOUT 端子から出力するパルスを設定します。
このコマンドの設定は相関軸で共有します。実行軸のどちらの軸に設定しても有効です。


<実行シーケンス>
COMMAND H'20
CP SPEC SET COMMAND
YSTATUS1 PORT ERROR = 0 ?
N
Y
① 相関軸 DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
YSTATUS1 PORT BUSY = 0 ?
② 相関軸 DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
XSTATUS1 PORT ERROR = 0 ?
③ 実行軸 DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
XSTATUS1 PORT BUSY = 0 ?
④ 実行軸 DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
DATA1 PORT WRITE
⑤ 実行軸 DRIVE DATA1 PORT に設定データを書き込みます。
COMMAND PORT WRITE
⑥ 実行軸 DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	–	–	–	–

D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	CPPOUT SEL2	CPPOUT SEL1	CPPOUT SEL0

●電源投入後の初期値は H'0 (アンダーライン側) です。

D0 ：CPPOUT SEL0
D1 ：CPPOUT SEL1
D2 ：CPPOUT SEL2

CPPOUT 端子から出力するパルスを選択します。

SEL2	SEL1	SEL0	CPPOUT から出力するパルス	
0	0	0	CPPIN 端子から入力するパルス	
0	0	1	メインチップ２軸補間ドライブの基本パルス	*1
0	1	0	X 軸の出力パルス（XOP）	
0	1	1	Y 軸の出力パルス（YOP）	
1	0	0	CPPIN 端子から入力するパルス	
1	0	1	メインチップ２軸補間ドライブの基本パルス	*1
1	1	0	X 軸のメイン軸補間ドライブの基本パルス	*1
1	1	1	Y 軸のメイン軸補間ドライブの基本パルス	*1

*1(メインチップ 2 軸相対アドレス補間関数で CPP STOP 機能使用時は、CPPOUT SEL=H'1 に設定してください)

*1：メイン軸補間ドライブまたはメインチップ２軸補間ドライブを実行するときに、
コマンド実行軸が発生する補間ドライブの基本パルスを出力します。
その他のドライブを実行する場合は、パルス出力なし（ハイレベル出力）になります。

D4 ：0

3-1-5. 直線補間ドライブの設定と実行

長軸と短軸の座標アドレスとドライブ仕様を指定して、相関 2 軸直線補間ドライブ または任意多軸直線補間ドライブを実行します。

・直線補間ドライブは実行軸の加減速パラメータでドライブします。
・直線補間ドライブでは、長軸パルスを補間ドライブの基本パルスとし、短軸側は長軸パルスを補間演算して補間パルスを出力します。
・現在位置を座標中心（0, 0）とした長軸と短軸の相対アドレスを、座標アドレスとします。
座標アドレスは、正数が ＋(CW)方向、負数が －(CCW)方向です。

※補間関数のメインチップ 2 軸相対アドレス直線補間ドライブ関数は、相関 2 軸直線補間ドライブを使用しています。

■直線補間ドライブ仕様

● DRIVE MODE
直線補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。

・直線補間 SCAN ドライブ
各補間軸は、長軸と補間軸のパルス比で、目的地の指定方向に補間パルス出力を続けます。
停止指令を検出すると、補間ドライブを終了します。

・直線補間 INDEX ドライブ
各補間軸は、長軸と補間軸のパルス比で、目的地の指定方向に補間パルス出力を続けます。
長軸パルスをカウントして、カウント数が長軸の目的地のパルス数になると、補間ドライブを終了します。

● CONST CP ENABLE
相関 2 軸直線補間ドライブとメイン軸直線補間ドライブで有効です。
線速一定制御を「無効にする／有効にする」を選択します。

・線速一定制御
直線補間ドライブしている２軸の合成速度を一定にする制御です。

コマンド実行軸が発生する補間ドライブの長軸パルスを線速一定制御します。
コマンド実行軸の長軸と短軸の２軸間で、２軸同時にパルス出力したときに、次の長軸パルスの出力周期を 1.414 倍にします。
線速一定で加減速ドライブを行うと、減速後の終了速度でのドライブが長くなります。

● CPP STOP ENABLE
相関 2 軸直線補間ドライブとメイン軸直線補間ドライブで有効です。
メイン軸の CPP STOP 機能を「有効にする／無効にする」を選択します。

・CPP STOP ENABLE = 1 を選択した場合は、CPPIN 端子に入力できるパルス速度の最高値は、2.5 MHz になります。

【メイン軸の CPP STOP 機能】
相関 2 軸直線補間ドライブとメイン軸直線補間ドライブ実行中に機能します。

・メイン軸の CPP STOP ENABLE = 1 にすると、メイン軸が発生する補間ドライブの基本パルスと CPPIN から入力するパルスを偏差カウントします。

・CPPIN のパルス数が、メイン軸の基本パルス数より２パルス分少なくなると、メイン軸のドライブを終了して、メイン軸が発生する補間ドライブの基本パルス出力を停止します。
相関 2 軸直線補間ドライブでは、両軸のドライブを終了して、基本パルス出力を停止します。

・CPP STOP 機能でドライブを終了した場合は、メイン軸のエラーになります。
ERROR STATUS の CPP STOP ERROR = 1 にします。

・メイン軸は CPP STOP 機能でドライブを終了しますが、サブ軸はドライブを終了しません。
メイン軸が CPP STOP 機能でドライブを終了した場合は、すべてのサブ軸に停止指令を実行して、ドライブを終了させてください。

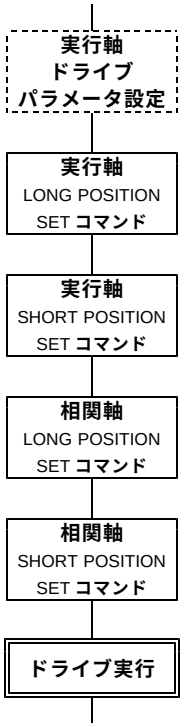
● CPP MASK ENABLE
サブ軸直線補間ドライブで有効です。
サブ軸の CPPIN マスク機能を「有効にする／無効にする」を選択します。

【サブ軸の CPP STOP 機能】
サブ軸補間ドライブ実行中に機能します。

- サブ軸の CPP MASK ENABLE = 1 にすると、サブ軸がエラー（DRIVE STATUS1 PORT の ERROR = 1）になると、CPPIN から入力するパルスをマスクします。
- CPPOUT SEL で CPPOUT 出力を「CPPIN から入力するパルス」に設定している場合は、CPPIN のマスクにより、CPPOUT 出力はハイレベル状態になります。
- CPPIN マスク機能で CPPIN をマスクした場合は、DRIVE STATUS5 PORT の CPP MASK = 1 になります。
  CPP MASK = 1 の間は、CPPIN のマスク状態を保持します。
  CPP MASK = 1 は、DRIVE STATUS1 PORT の ERROR = 1 → 0 で CPP MASK = 0 になります。
- AL-Ⅱシリーズでは、相関 2 軸の MCC 間同士を CPPIN と CPPOUT をデジーチェーン接続しています。
  CPPIN と CPPOUT をデジーチェーン接続した多軸直線補間ドライブで、CPP STOP 機能と CPPIN マスク機能を有効にすると、サブ軸にエラーが発生した場合にすべての補間軸のパルス出力を停止させることができます。
- また、多軸直線補間ドライブの補間軸を、メイン軸補間ドライブまたはメインチップ２軸補間ドライブで構成して CPP STOP 機能を有効にすると、１軸が停止指令またはエラーにより補間ドライブを終了した場合に、すべての補間ドライブを終了させることができます。

■相関 2 軸直線補間ドライブの実行シーケンス

相関軸の 2 軸に実行する直線補間ドライブです。



①実行軸に加減速ドライブに必要なパラメータを設定します。
・第一パルス周期
・加減速パラメータ

②実行軸に、長軸の座標アドレスを設定します。

③実行軸に短軸の座標アドレスを設定します。

④相関軸に長軸の座標アドレスを設定します。
(設定するデータは実行軸と同じデータです。)

⑤相関軸に短軸の座標アドレスを設定します。
(設定するデータは実行軸と同じデータです。)

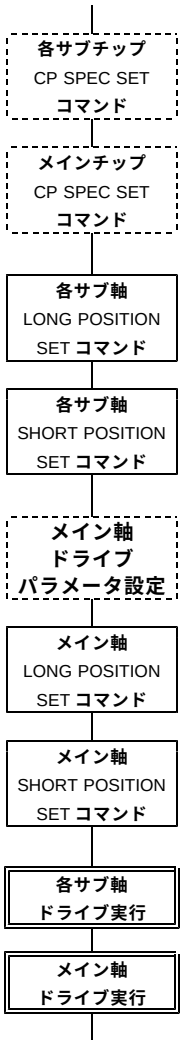
⑥実行軸にドライブ仕様を指定して、ドライブを実行します。

初期値および設定値に対して変更が必要な場合に設定します。

・相関 2 軸直線補間ドライブでは、CP SPEC SET による CPPOUT の設定は不要です。

■任意多軸直線補間ドライブの実行シーケンス

任意軸間、任意軸数で実行する直線補間ドライブです。



①各サブ軸のチップに、CPPOUT 出力を設定します。
CPPOUT SEL = "CPPIN から入力するパルス" に設定します。

②メイン軸のチップに、CPPOUT 出力を設定します。
CPPOUT SEL = "メイン軸の補間ドライブの基本パルス" に設定します。

③各サブ軸に、長軸の座標アドレスを設定します。

④各サブ軸に、短軸の座標アドレスを設定します。

⑤メイン軸に加減速ドライブに必要なパラメータを設定します。
・第一パルス周期
・加減速パラメータ

⑥メイン軸に、長軸の座標アドレスを設定します。

⑦メイン軸に、短軸の座標アドレスを設定します。

⑧各サブ軸に、サブ軸直線補間ドライブを実行します。

⑨メイン軸に、メイン軸直線補間ドライブを実行します。

初期値および設定値に対して変更が必要な場合に設定します。

・サブ軸直線補間ドライブを実行すると、ドライブがスタンバイ状態になります。
メイン軸直線補間ドライブを実行すると、長軸パルスを出力して、ドライブを開始します。

・各サブ軸は CPPIN 端子から入力するパルスを長軸パルスとして、ドライブを開始します。

(1) LONG POSITION SET

直線補間ドライブの、長軸の座標アドレスを設定します。

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A15	←				長軸の目的地の座標アドレス									→	A0

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A31	←				長軸の目的地の座標アドレス									→	A16

●電源投入後の初期値は H'0000_0000 です。

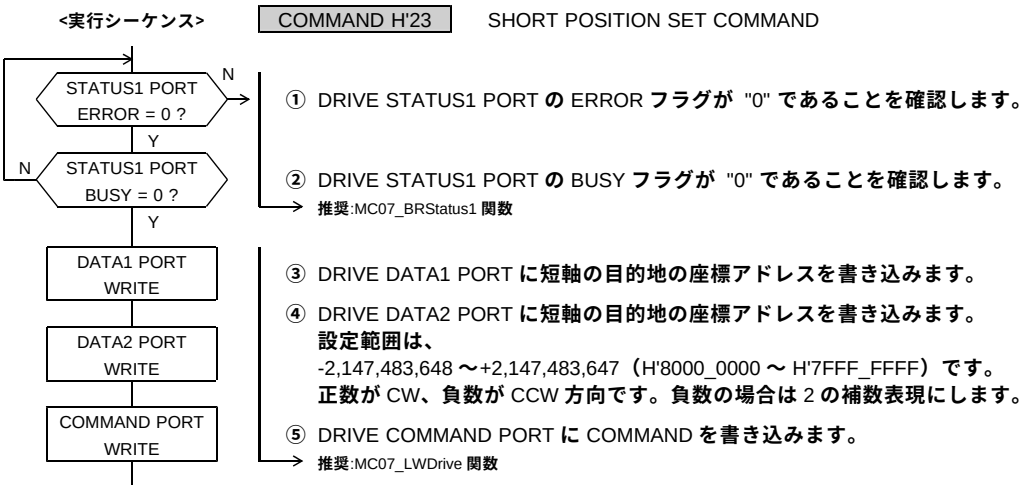
指定する座標アドレスは、現在位置を座標中心（0, 0）とした相対アドレスです。

「長軸の目的地の座標アドレス」には、
各補間軸の中で補間パルス数が大きい補間軸（長軸）の目的地を設定します。

ドライブ実行コマンドの PULSE SEL で指定した軸の座標アドレスの符号が、
出力する補間パルスの動作方向になります。

(2) SHORT POSITION SET

直線補間ドライブの、短軸の座標アドレスを設定します。



DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A15	←					短軸の目的地の座標アドレス								→	A0

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A31	←					短軸の目的地の座標アドレス								→	A16

●電源投入後の初期値は H'0000_0000 です。

指定する座標アドレスは、現在位置を座標中心（0, 0）とした相対アドレスです。

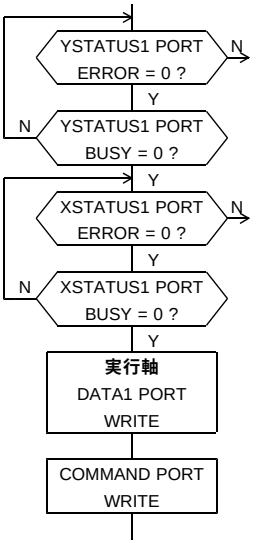
「短軸の目的地の座標アドレス」には、短軸の目的地（符号付きアドレス）を設定します。
・「長軸の座標アドレスの絶対値≧短軸の座標アドレスの絶対値」に設定します。

ドライブ実行コマンドの PULSE SEL で指定した軸（長軸／短軸）の座標アドレスの符号が、
出力する補間パルスの動作方向になります。

(3) MAIN XY STRAIGHT CP

2 軸相関コマンドです。相関軸両軸が BUSY = 0, ERROR = 0 のときにコマンドを実行します。
メインチップの相関 2 軸直線補間ドライブを実行します。相関軸のどちらの軸に実行しても有効です。
直線補間ドライブは実行軸の加減速パラメータで動作します。

<実行シーケンス>



MAIN XY STRAIGHT CP COMMAND

① 相関軸 DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。

② 相関軸 DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数

③ 実行軸 DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。

④ 実行軸 DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数

⑤ 実行軸 DRIVE DATA1 PORT に設定データを書き込みます。

⑥ 実行軸 DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	–	–	–	–

D7	D6	D5	D4	D3	D2	D1	D0
–	–	YPULSE SEL	XPULSE SEL	–	0	CONST CP ENABLE	DRIVE MODE

D0 ：DRIVE MODE
直線補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。
0 ：連続ドライブにする (SCAN ドライブ)
1 ：位置決めドライブにする (INDEX ドライブ)

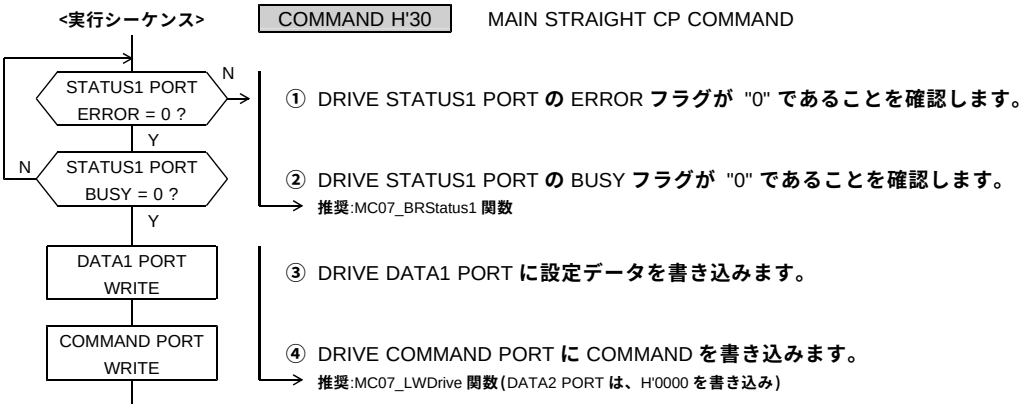
D1 ：CONST CP ENABLE
線速一定制御を「無効にする／有効にする」を選択します。
0 ：線速一定制御を無効にする
1 ：線速一定制御を有効にする

D4 ：XPULSE SEL
相関軸の内、X 軸(または Z 軸)に出力する補間パルスを選択します。
0 ：X 軸に LONG POSITION （長軸）の補間パルスを出力する
1 ：X 軸に SHORT POSITION （短軸）の補間パルスを出力する

D5 ：YPULSE SEL
相関軸の内、Y 軸(または A 軸)に出力する補間パルスを選択します。
0 ：Y 軸に LONG POSITION （長軸）の補間パルスを出力する
1 ：Y 軸に SHORT POSITION （短軸）の補間パルスを出力する

(4) MAIN STRAIGHT CP

1 軸単位で直線補間ドライブを行うコマンドです。
任意軸間の直線補間、または複数軸で直線補間ドライブさせるときにメイン軸に実行します。
メイン軸の直線補間ドライブは実行軸の加減速パラメータで動作します。


<実行シーケンス>
COMMAND H'30
MAIN STRAIGHT CP COMMAND
STATUS1 PORT ERROR = 0 ?
N
Y
STATUS1 PORT BUSY = 0 ?
N
Y
DATA1 PORT WRITE
COMMAND PORT WRITE
① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
③ DRIVE DATA1 PORT に設定データを書き込みます。
④ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	–	–	–	–

D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	PULSE SEL	–	CPP STOP ENABLE	CONST CP ENABLE	DRIVE MODE

D0 ：DRIVE MODE
直線補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。
0 ：連続ドライブにする (SCAN ドライブ)
1 ：位置決めドライブにする (INDEX ドライブ)

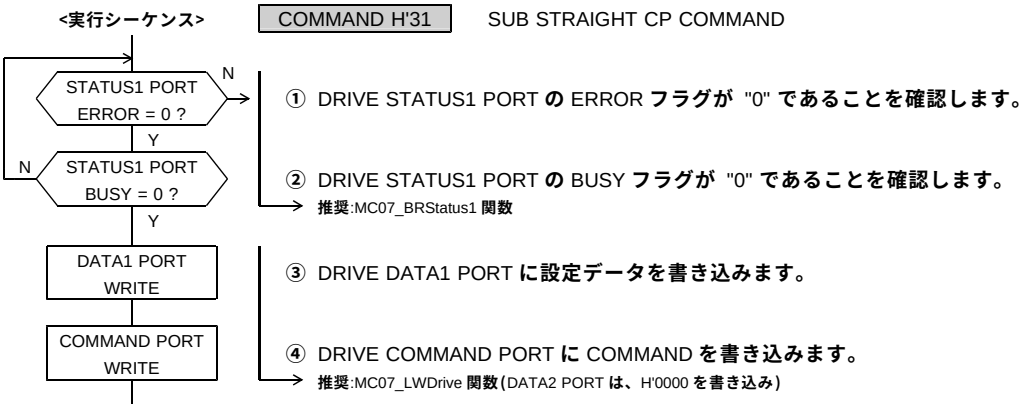
D1 ：CONST CP ENABLE
線速一定制御を「無効にする／有効にする」を選択します。
0 ：線速一定制御を無効にする
1 ：線速一定制御を有効にする

D2 ：CPP STOP ENABLE
CPP STOP 機能を「有効にする／無効にする」を選択します。
0 ：CPP STOP 機能を無効にする
1 ：CPP STOP 機能を有効にする

D4 ：PULSE SEL
出力する補間パルスを選択します。
0 ：LONG POSITION（長軸）の補間パルスを出力する
1 ：SHORT POSITION（短軸）の補間パルスを出力する

(5) SUB STRAIGHT CP

1 軸単位で直線補間ドライブを行うコマンドです。
任意軸間の直線補間、または複数軸で直線補間ドライブさせるときにサブ軸に実行します。
サブ軸の直線補間ドライブは CPPIN 入力パルスで動作します。


<実行シーケンス>
COMMAND H'31
SUB STRAIGHT CP COMMAND
STATUS1 PORT ERROR = 0 ?
N
Y
STATUS1 PORT BUSY = 0 ?
N
Y
DATA1 PORT WRITE
COMMAND PORT WRITE
① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
③ DRIVE DATA1 PORT に設定データを書き込みます。
④ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	–	–	–	–

D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	PULSE SEL	–	CPP MASK ENABLE	–	DRIVE MODE

D0 ：DRIVE MODE
直線補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。
0 ：連続ドライブにする (SCAN ドライブ)
1 ：位置決めドライブにする (INDEX ドライブ)

D2 ：CPP MASK ENABLE
CPPIN マスク機能を「有効にする／無効にする」を選択します。
0 ：CPPIN マスク機能を無効にする
1 ：CPPIN マスク機能を有効にする

D4 ：PULSE SEL
出力する補間パルスを選択します。
0 ：LONG POSITION（長軸）の補間パルスを出力する
1 ：SHORT POSITION（短軸）の補間パルスを出力する

3-1-6. 円弧補間ドライブの設定と実行

現在位置の X-Y 座標アドレスと、目的地の短軸座標までの短軸パルス数と、ドライブ仕様を指定して、
相関 2 軸円弧補間ドライブ および 任意 2 軸直線補間ドライブを実行します。

・円弧補間ドライブは実行軸の加減速パラメータをドライブします。
・円弧補間ドライブでは、円弧の中心座標からみた短軸側の短軸パルスを補間ドライブの基本パルスとし、
長軸側は短軸パルスを補間演算して補間パルスを出力します。
・円弧の中心点座標を（0, 0）とした X 軸と Y 軸の相対アドレスを、X-Y 座標アドレスとします。
座標アドレスは、正数が ＋(CW)方向、負数が －(CCW)方向です。

※補間関数のメインチップ 2 軸相対アドレス円弧補間ドライブ関数は相関 2 軸円弧補間ドライブを
使用しています。

■円弧補間ドライブ仕様

● DRIVE MODE
円弧補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。

・円弧補間 SCAN ドライブ
各補間軸は、指定の円弧半径と回転方向で、補間パルス出力を続けます。
停止指令を検出すると、補間ドライブを終了します。

・円弧補間 INDEX ドライブ
各補間軸は、指定の円弧半径と回転方向で、補間パルス出力を続けます。
短軸パルスをカウントして、カウント数が指定の短軸パルス数になると、補間ドライブを終了します。

● CONST CP ENABLE
相関 2 軸円弧補間ドライブとメイン軸円弧補間ドライブで有効です。
線速一定制御を「無効にする／有効にする」を選択します。

・線速一定制御
補間ドライブしている２軸の合成速度を一定にする制御です。

コマンド実行軸が発生する補間ドライブの短軸パルスを線速一定制御します。
X 座標軸と Y 座標軸の２軸間で、２軸同時にパルス出力したときに、次の短軸パルスの出力周期を
1.414 倍にします。
線速一定で加減速ドライブを行うと、減速後の終了速度でのドライブが長くなります。

● CPP STOP ENABLE
2 軸相関円弧補間ドライブとメイン軸円弧補間ドライブで有効です。
メイン軸の CPP STOP 機能を「有効にする／無効にする」を選択します。

・CPP STOP ENABLE = 1 を選択した場合は、CPPIN 端子に入力できるパルス速度の最高値は、
2.5 MHz になります。

【メイン軸の CPP STOP 機能】
2 軸相関円弧補間ドライブとメイン軸円弧補間ドライブ実行中に機能します。

・メイン軸の CPP STOP ENABLE = 1 にすると、メイン軸が発生する補間ドライブの基本パルスと
CPPIN から入力するパルスを偏差カウントします。

・CPPIN のパルス数が、メイン軸の基本パルス数より２パルス分少なくなると、メイン軸のドライブ
を終了して、メイン軸が発生する補間ドライブの基本パルス出力を停止します。
２軸相関円弧補間ドライブでは、両軸のドライブを終了して、基本パルス出力を停止します。

・CPP STOP 機能でドライブを終了した場合は、メイン軸のエラーになります。
ERROR STATUS の CPP STOP ERROR = 1 にします。

・メイン軸は CPP STOP 機能でドライブを終了しますが、サブ軸はドライブを終了しません。
メイン軸が CPP STOP 機能でドライブを終了した場合は、すべてのサブ軸に停止指令を実行して、
ドライブを終了させてください。

● CPP MASK ENABLE

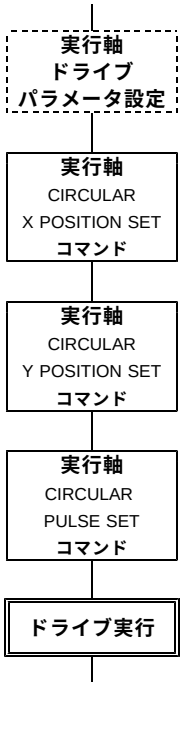
サブ軸円弧補間ドライブで有効です。
サブ軸の CPPIN マスク機能を「有効にする／無効にする」を選択します。

【サブ軸の CPP STOP 機能】
サブ軸補間ドライブ実行中に機能します。

- サブ軸の CPP MASK ENABLE = 1 にすると、サブ軸がエラー（DRIVE STATUS1 PORT の ERROR = 1）になると、CPPIN から入力するパルスをマスクします。
- CPPOUT SEL で CPPOUT 出力を「CPPIN から入力するパルス」に設定している場合は、CPPIN のマスクにより、CPPOUT 出力はハイレベル状態になります。
- CPPIN マスク機能で CPPIN をマスクした場合は、DRIVE STATUS5 PORT の CPP MASK = 1 になります。
  CPP MASK = 1 の間は、CPPIN のマスク状態を保持します。
  CPP MASK = 1 は、DRIVE STATUS1 PORT の ERROR = 1 → 0 で CPP MASK = 0 になります。
- AL-Ⅱシリーズでは、相関 2 軸の MCC 間同士を CPPIN と CPPOUT をデジーチェーン接続しています。
  任意軸円弧補間ドライブで、CPP STOP 機能と CPPIN マスク機能を有効にすると、サブ軸にエラーが発生した場合にすべての補間軸のパルス出力を停止させることができます。
- また、任意軸円弧補間ドライブの補間軸を、メイン軸補間ドライブまたは２軸相関円弧補間ドライブで構成して CPP STOP 機能を有効にすると、１軸が停止指令またはエラーにより補間ドライブを終了した場合に、すべての補間ドライブを終了させることができます。

■相関 2 軸円弧補間ドライブの実行シーケンス

相関軸の 2 軸に実行する円弧補間ドライブです。



①実行軸にドライブに必要なパラメータを設定します。
・第一パルス周期
・加減速パラメータ

②実行軸に現在位置の X 座標アドレスを設定します。

③実行軸に現在位置の Y 座標アドレスを設定します。

④実行軸に目的地の短軸座標までのパルス数を設定します。

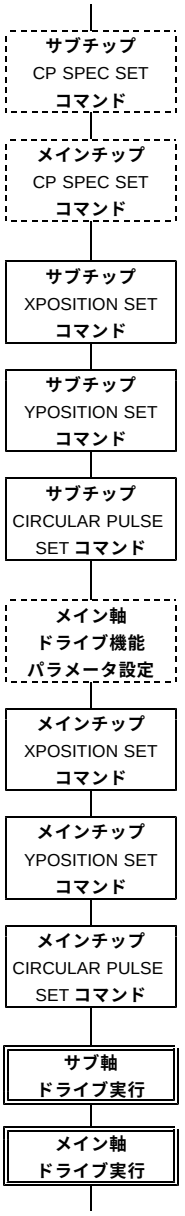
⑤実行軸にドライブ仕様を指定して、ドライブを実行します。

初期値および設定値に対して変更が必要な場合に設定します。

・相関 2 軸円弧補間ドライブでは、CP SPEC SET による CPPOUT の設定は不要です。

■任意 2 軸円弧補間ドライブの実行シーケンス

任意 2 軸間(2 チップ間)で実行する円弧補間ドライブです。



① サブ軸のチップに、CPPOUT 出力を設定します。
CPPOUT SEL = "CPPIN から入力するパルス" に設定します。

② メイン軸のチップに、CPPOUT 出力を設定します。
CPPOUT SEL = "メイン軸の補間ドライブの基本パルス" に設定します。

③ サブ軸のチップに、現在位置の X 座標アドレスを設定します。

④ サブ軸のチップに、現在位置の Y 座標アドレスを設定します。

⑤ サブ軸のチップに、目的地の短軸座標までのパルス数を設定します。

⑥ メイン軸に、加減速ドライブに必要なパラメータを設定します。

⑦ メイン軸のチップに、現在位置の X 座標アドレスを設定します。

⑧ メイン軸のチップに、現在位置の Y 座標アドレスを設定します。

⑨ メイン軸のチップに、目的地の短軸座標までのパルス数を設定します。

⑩ サブ軸に、サブ軸円弧補間ドライブを実行します。

⑪ メイン軸に、メイン軸円弧補間ドライブを実行します。

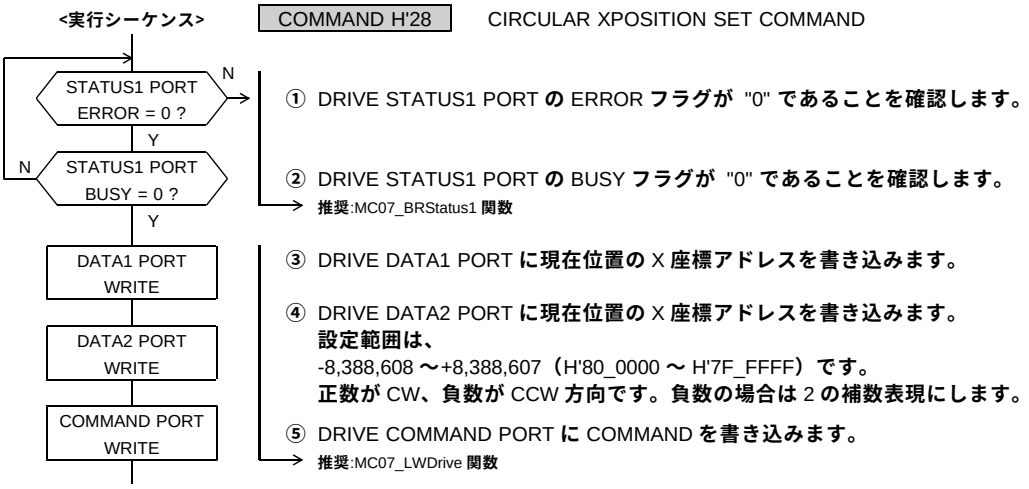
初期値および設定値に対して変更が必要な場合に設定します。

・サブ軸円弧補間ドライブを実行すると、ドライブがスタンバイ状態になります。

・メイン軸円弧補間ドライブを実行すると、短軸パルスを出力して、ドライブを開始します。
サブ軸は CPPIN 端子から入力するパルスを短軸パルスとして、ドライブを開始します。

(1) CIRCULAR XPOSITION SET

円弧の中心点座標を（0, 0）とした現在位置の X 座標アドレスを設定します。
このコマンドの設定は相関軸で共有します。
実行軸のどちらの軸に設定しても有効です。


<実行シーケンス>
COMMAND H'28
CIRCULAR XPOSITION SET COMMAND
STATUS1 PORT ERROR = 0 ?
N
Y
STATUS1 PORT BUSY = 0 ?
N
Y
DATA1 PORT WRITE
DATA2 PORT WRITE
COMMAND PORT WRITE
① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
③ DRIVE DATA1 PORT に現在位置の X 座標アドレスを書き込みます。
④ DRIVE DATA2 PORT に現在位置の X 座標アドレスを書き込みます。
設定範囲は、
-8,388,608 ～+8,388,607（H'80_0000 ～ H'7F_FFFF）です。
正数が CW、負数が CCW 方向です。負数の場合は 2 の補数表現にします。
⑤ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A15	←					現在位置の X 座標アドレス								→	A0

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
−	−	−	−	−	−	−	−	A23	←	現在位置の X 座標アドレス				→	A16

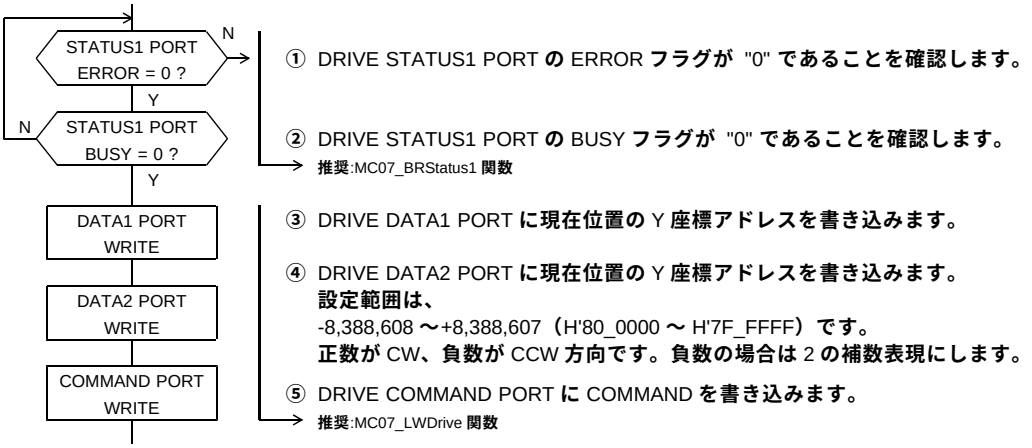
●電源投入後の初期値は H'00_0000 です。

・指定する座標アドレスは、円弧の中心点座標を（0, 0）とした X 軸の相対アドレスです。

(2) CIRCULAR YPOSITION SET

円弧の中心点座標を（0, 0）とした現在位置の Y 座標アドレスを設定します。
このコマンドの設定は相関軸で共有します。
実行軸のどちらの軸に設定しても有効です。

<実行シーケンス>　COMMAND H'29　CIRCULAR YPOSITION SET COMMAND


STATUS1 PORT ERROR = 0 ?
N
Y
STATUS1 PORT BUSY = 0 ?
N
Y
DATA1 PORT WRITE
DATA2 PORT WRITE
COMMAND PORT WRITE
① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
③ DRIVE DATA1 PORT に現在位置の Y 座標アドレスを書き込みます。
④ DRIVE DATA2 PORT に現在位置の Y 座標アドレスを書き込みます。
設定範囲は、
-8,388,608 ～+8,388,607（H'80_0000 ～ H'7F_FFFF）です。
正数が CW、負数が CCW 方向です。負数の場合は 2 の補数表現にします。
⑤ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A15	←				現在位置の Y 座標アドレス									→	A0

DRIVE DATA2 PORT の設定データ

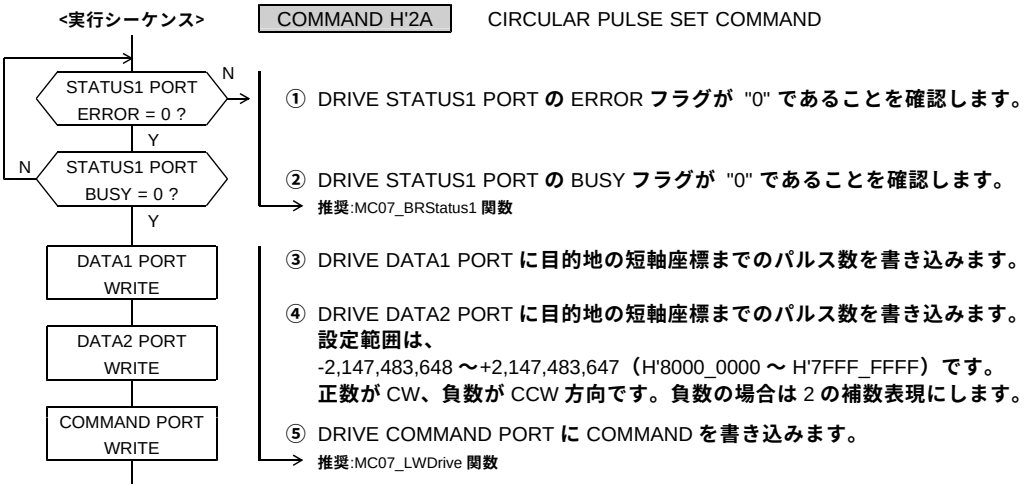
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	A23	← 現在位置の Y 座標アドレス					→	A16

●電源投入後の初期値は H'00_0000 です。

・指定する座標アドレスは、円弧の中心点座標を（0, 0）とした Y 軸の相対アドレスです。

(3) CIRCULAR PULSE SET

現在位置の X-Y 座標アドレスから目的地の短軸座標までの短軸パルス数を設定します。
このコマンドの設定は相関軸で共有します。
実行軸のどちらの軸に設定しても有効です。



DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15 ←					目的地の短軸座標までのパルス数									→	D0

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31 ←					目的地の短軸座標までのパルス数									→	D16

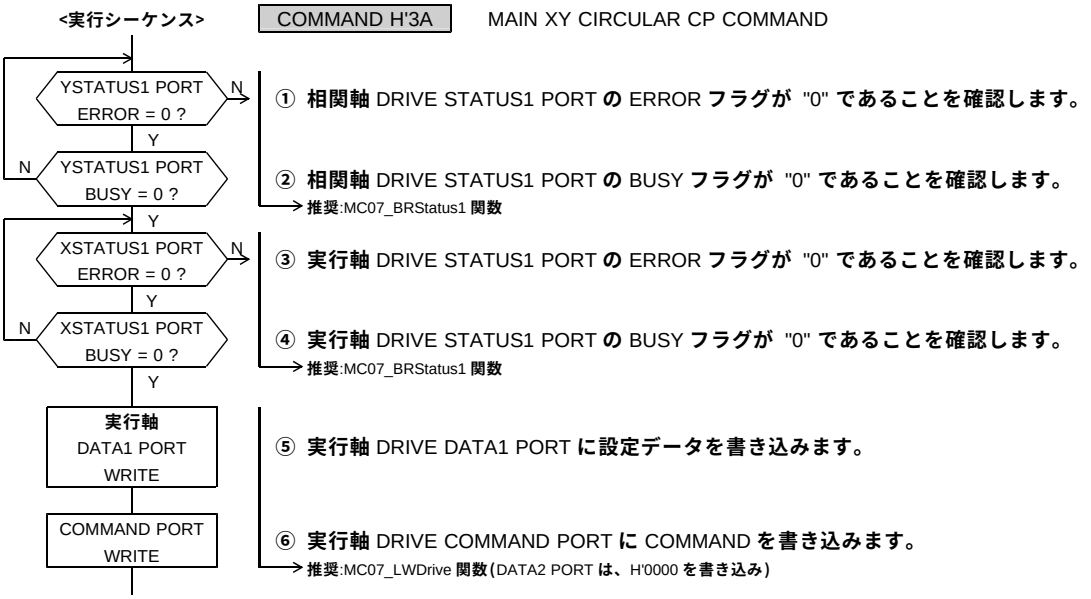
●電源投入後の初期値は H'0000_0000（0 パルス）です。

指定するパルス数は、
目的地の短軸座標に到達するまでに経由する、各象限の短軸の合計パルス数です。
・短軸パルス数の計算式は、「4-1-5.(4) 円弧補間ドライブ」をご覧ください。
デバイスドライバの円弧補間短軸 PULSE 数ゲット関数を使用すると
指定された円弧の中心点相対アドレス、目的地相対アドレス、回転方向をもとに
目的地の短軸座標までのパルス数を算出します。

円弧を描く回転方向は、パルス数の符号で指定します。
・正数を指定すると CW 方向に回転します。
・負数を指定すると CCW 方向に回転します。

(4) MAIN XY CIRCULAR CP

2 軸相関コマンドです。相関軸両軸が BUSY = 0, ERROR = 0 のときにコマンドを実行します。
メインチップの相関 2 軸円弧補間ドライブを実行します。相関軸のどちらの軸に実行しても有効です。
円弧補間ドライブは実行軸の加減速パラメータで動作します。


<実行シーケンス>
COMMAND H'3A
MAIN XY CIRCULAR CP COMMAND
YSTATUS1 PORT ERROR = 0 ?
N
Y
① 相関軸 DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
YSTATUS1 PORT BUSY = 0 ?
② 相関軸 DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
XSTATUS1 PORT ERROR = 0 ?
③ 実行軸 DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
XSTATUS1 PORT BUSY = 0 ?
④ 実行軸 DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
実行軸 DATA1 PORT WRITE
⑤ 実行軸 DRIVE DATA1 PORT に設定データを書き込みます。
COMMAND PORT WRITE
⑥ 実行軸 DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	–	–	–	–

D7	D6	D5	D4	D3	D2	D1	D0
–	–	YPULSE SEL	XPULSE SEL	–	0	CONST CP ENABLE	DRIVE MODE

D0 ：DRIVE MODE
円弧補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。
0 ：連続ドライブにする (SCAN ドライブ)
1 ：位置決めドライブにする (INDEX ドライブ)

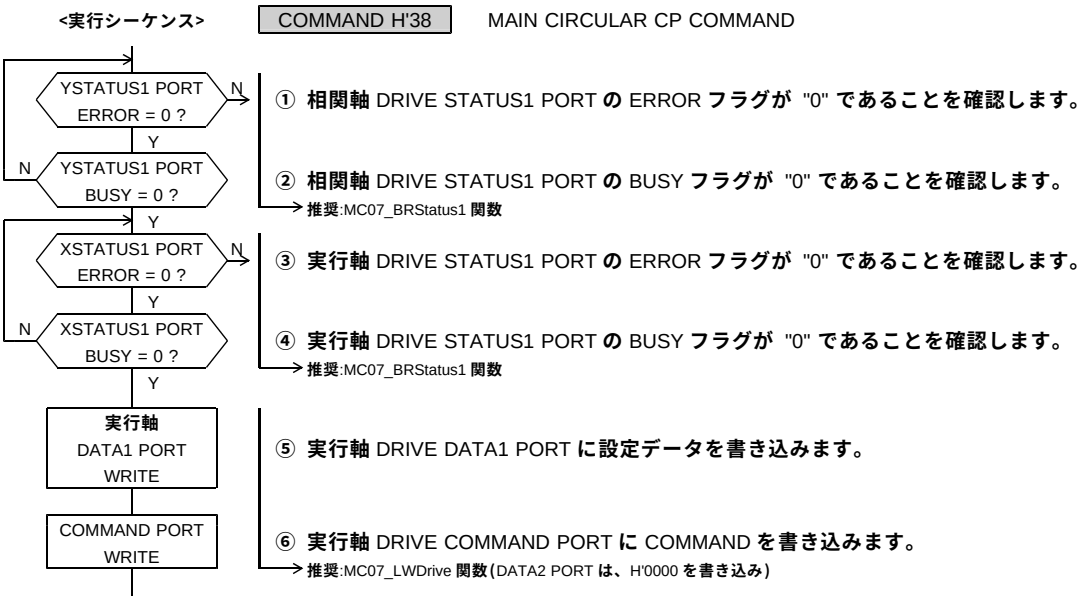
D1 ：CONST CP ENABLE
線速一定制御を「無効にする／有効にする」を選択します。
0 ：線速一定制御を無効にする
1 ：線速一定制御を有効にする

D4 ：XPULSE SEL
相関軸の内、X 軸(または Z 軸)に出力する補間パルスを選択します。
0 ：X 軸に円弧補間演算の X 座標アドレスの補間パルス（XCP）を出力する
1 ：X 軸に円弧補間演算の Y 座標アドレスの補間パルス（YCP）を出力する

D5 ：YPULSE SEL
相関軸の内、Y 軸(または A 軸)に出力する補間パルスを選択します。
0 ：Y 軸に円弧補間演算の X 座標アドレスの補間パルス（XCP）を出力する
1 ：Y 軸に円弧補間演算の Y 座標アドレスの補間パルス（YCP）を出力する

(5) MAIN CIRCULAR CP

2 軸相関コマンドです。X, Y 軸が BUSY = 0, ERROR = 0 のときにコマンドを実行します。
1 軸単位で円弧補間ドライブを行うコマンドです。
任意軸間の円弧補間ドライブさせるときにメイン軸に実行します。
メイン軸の円弧補間ドライブは実行軸の加減速パラメータで動作します。


<実行シーケンス>
COMMAND H'38
MAIN CIRCULAR CP COMMAND
YSTATUS1 PORT ERROR = 0 ?
N
Y
① 相関軸 DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
YSTATUS1 PORT BUSY = 0 ?
② 相関軸 DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
XSTATUS1 PORT ERROR = 0 ?
③ 実行軸 DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
XSTATUS1 PORT BUSY = 0 ?
④ 実行軸 DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
実行軸 DATA1 PORT WRITE
⑤ 実行軸 DRIVE DATA1 PORT に設定データを書き込みます。
COMMAND PORT WRITE
⑥ 実行軸 DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	–	–	–	–

D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	PULSE SEL	–	CPP STOP ENABLE	CONST CP ENABLE	DRIVE MODE

D0 ：DRIVE MODE
円弧補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。
0 ：連続ドライブにする (SCAN ドライブ)
1 ：位置決めドライブにする (INDEX ドライブ)

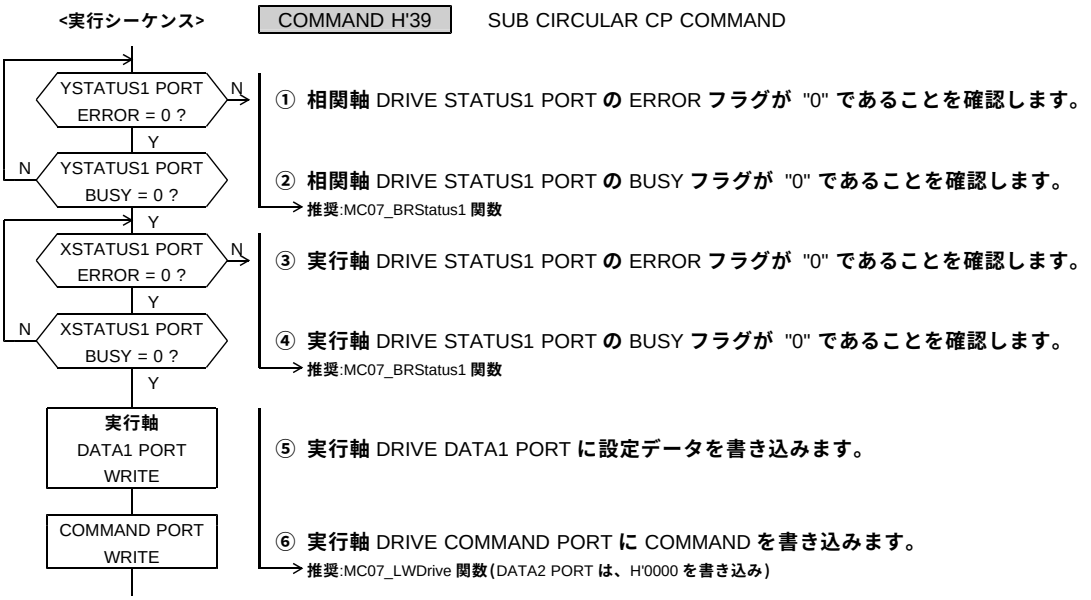
D1 ：CONST CP ENABLE
線速一定制御を「無効にする／有効にする」を選択します。
0 ：線速一定制御を無効にする
1 ：線速一定制御を有効にする

D2 ：CPP STOP ENABLE
CPP STOP 機能を「有効にする／無効にする」を選択します。
0 ：CPP STOP 機能を無効にする
1 ：CPP STOP 機能を有効にする

D4 ：PULSE SEL
出力する補間パルスを選択します。
0 ：円弧補間演算の X 座標アドレスの補間パルス（XCP）を出力する
1 ：円弧補間演算の Y 座標アドレスの補間パルス（YCP）を出力する

(6) SUB CIRCULAR CP

2 軸相関コマンドです。X, Y 軸が BUSY = 0, ERROR = 0 のときにコマンドを実行します。
1 軸単位で円弧補間ドライブを行うコマンドです。
任意軸間の円弧補間ドライブさせるときにサブ軸に実行します。
サブ軸の円弧補間ドライブは CPPIN 入力パルスで動作します。


<実行シーケンス>
COMMAND H'39
SUB CIRCULAR CP COMMAND
YSTATUS1 PORT ERROR = 0 ?
N
Y
YSTATUS1 PORT BUSY = 0 ?
① 相関軸 DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
② 相関軸 DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
XSTATUS1 PORT ERROR = 0 ?
XSTATUS1 PORT BUSY = 0 ?
③ 実行軸 DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。
④ 実行軸 DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。
推奨:MC07_BRStatus1 関数
実行軸 DATA1 PORT WRITE
⑤ 実行軸 DRIVE DATA1 PORT に設定データを書き込みます。
COMMAND PORT WRITE
⑥ 実行軸 DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	–	–	–	–

D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	PULSE SEL	–	CPP MASK ENABLE	–	DRIVE MODE

D0 ：DRIVE MODE
円弧補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。
0 ：連続ドライブにする (SCAN ドライブ)
1 ：位置決めドライブにする (INDEX ドライブ)

D2 ：CPP MASK ENABLE
CPPIN マスク機能を「有効にする／無効にする」を選択します。
0 ：CPPIN マスク機能を無効にする
1 ：CPPIN マスク機能を有効にする

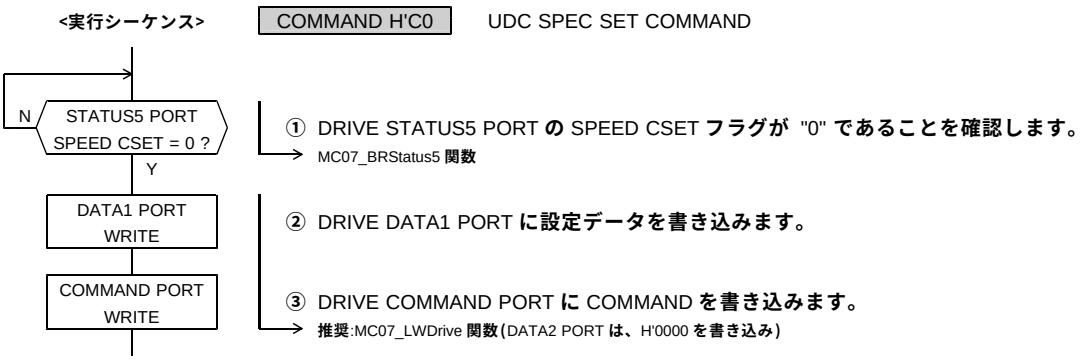
D4 ：PULSE SEL
出力する補間パルスを選択します。
0 ：円弧補間演算の X 座標アドレスの補間パルス（XCP）を出力する
1 ：円弧補間演算の Y 座標アドレスの補間パルス（YCP）を出力する

3-1-7. UP/DOWN/CONST ドライブ CHANGE の設定と実行

変更動作点を設定して、UP/DOWN/CONST のドライブ CHANGE を実行します。
変更動作点の設定は、変更動作点の変更が必要な場合に設定します。

(1) UDC SPEC SET

UP/DOWN/CONST のドライブ CHANGE 指令を実行する変更動作点を設定します。



DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	–	CONST TYPE2	CONST TYPE1	CONST TYPE0

D7	D6	D5	D4	D3	D2	D1	D0
–	DOWN TYPE2	DOWN TYPE1	DOWN TYPE0	–	UP TYPE2	UP TYPE1	UP TYPE0

●電源投入後の初期値は H'000 (アンダーライン側) です。

D2--D0 ：UP TYPE2--0
UP DRIVE コマンドのドライブ CHANGE 指令を実行する変更動作点を選択します。

D6--D4 ：DOWN TYPE2--0
DOWN DRIVE コマンドのドライブ CHANGE 指令を実行する変更動作点を選択します。

D10--D8 ：CONST TYPE2--0
CONST DRIVE コマンドのドライブ CHANGE 指令を実行する変更動作点を選択します。

TYPE2	TYPE1	TYPE0	ドライブ CHANGE を実行する変更動作点 <レベル検出>
0	0	0	ドライブ CHANGE コマンドの書き込みで実行する
0	0	1	設定禁止
0	1	0	STATUS5 PORT の SS0 = 1 で実行する (注)
0	1	1	設定禁止
1	0	0	設定禁止
1	0	1	設定禁止
1	1	0	設定禁止
1	1	1	設定禁止

(注) SS0 信号は SPEC INITIALIZE2 コマンドで「各種トリガ」に設定している場合に有効です。

・SS0 信号は下記の軸に対応しています。Z 軸および A 軸には対応していません。

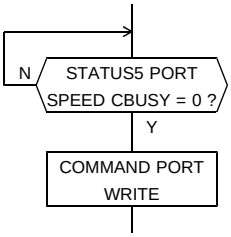
/IN0 信号	X 軸の SS0 信号
/IN1 信号	Y 軸の SS0 信号

(2) UP DRIVE

実行中のパルス出力を、最高速度まで加速または減速します。

<実行シーケンス>

COMMAND H'C4　UP DRIVE COMMAND



① DRIVE STATUS5 PORT の SPEED CBUSY フラグが "0" であることを確認します。
→ MC07_BRStatus5 関数

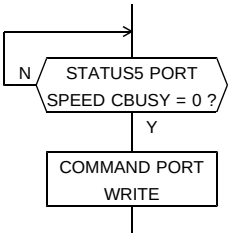
② DRIVE COMMAND PORT に COMMAND を書き込みます。
→ 推奨:MC07_BWDriveCommand 関数

(3) DOWN DRIVE

実行中のパルス出力を、終了速度まで加速または減速します。

<実行シーケンス>

COMMAND H'C5　DOWN DRIVE COMMAND



① DRIVE STATUS5 PORT の SPEED CBUSY フラグが "0" であることを確認します。
→ MC07_BRStatus5 関数

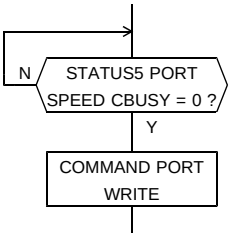
② DRIVE COMMAND PORT に COMMAND を書き込みます。
→ 推奨:MC07_BWDriveCommand 関数

(4) CONST DRIVE

実行中の加速または減速を終了して、パルス出力を一定速にします。

<実行シーケンス>

COMMAND H'C6　CONST DRIVE COMMAND



① DRIVE STATUS5 PORT の SPEED CBUSY フラグが "0" であることを確認します。
→ MC07_BRStatus5 関数

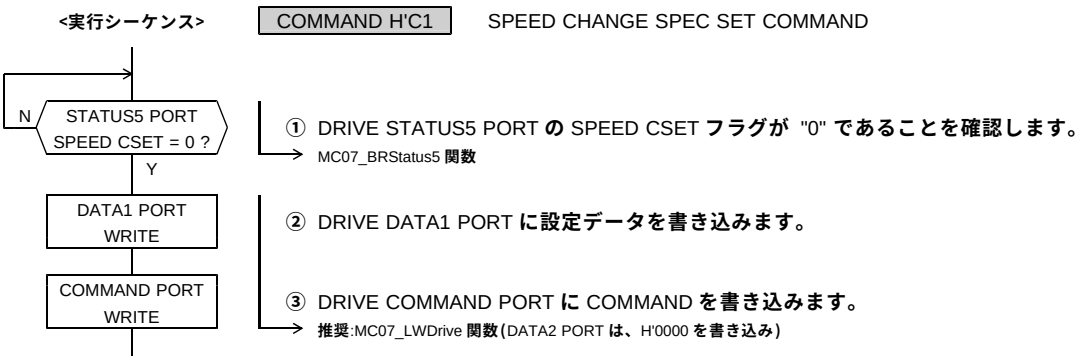
② DRIVE COMMAND PORT に COMMAND を書き込みます。
→ 推奨:MC07_BWDriveCommand 関数

3-1-8. SPEED CHANGE の設定と実行

変更動作点を設定して、SPEED CHANGE を実行します。
変更動作点の設定は、変更動作点の変更が必要な場合に設定します。

(1) SPEED CHANGE SPEC SET

SPEED CHANGE 指令を実行する変更動作点を設定します。



DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	–	–	–	–

D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	SPEED CHANGE TYPE2	SPEED CHANGE TYPE1	SPEED CHANGE TYPE0

●電源投入後の初期値は H'0 (アンダーライン側) です。

D2--D0 ：SPEED CHANGE TYPE2--0
SPEED CHANGE 指令を実行する変更動作点を選択します。

TYPE2	TYPE1	TYPE0	SPEED CHANGE を実行する変更動作点 <レベル検出>
0	0	0	SPEED CHANGE コマンドの書き込みで実行する
0	0	1	設定禁止
0	1	0	STATUS5 PORT の SS0 = 1 で実行する (注)
0	1	1	設定禁止
1	0	0	設定禁止
1	0	1	設定禁止
1	1	0	設定禁止
1	1	1	設定禁止

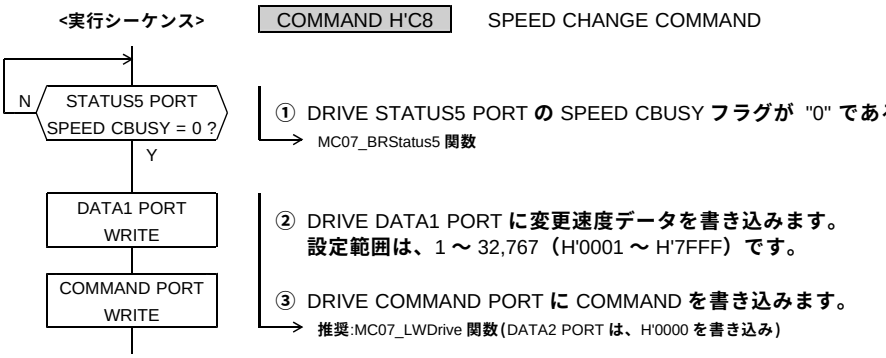
(注) SS0 信号は SPEC INITIALIZE2 コマンドで「各種トリガ」に設定している場合に有効です。

・SS0 信号は下記の軸に対応しています。Z 軸および A 軸には対応していません。

/IN0 信号	X 軸の SS0 信号
/IN1 信号	Y 軸の SS0 信号

(2) SPEED CHANGE

実行中のパルス出力を、指定したドライブパルス速度まで加速または減速します。



DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
−	D14 ←					SPEED CHANGE データ								→	D0

SPEED CHANGE データの設定値が "0" の場合は、"1" に補正します。
・SPEED CHANGE の速度（Hz）＝ SPEED CHANGE データ x RESOL

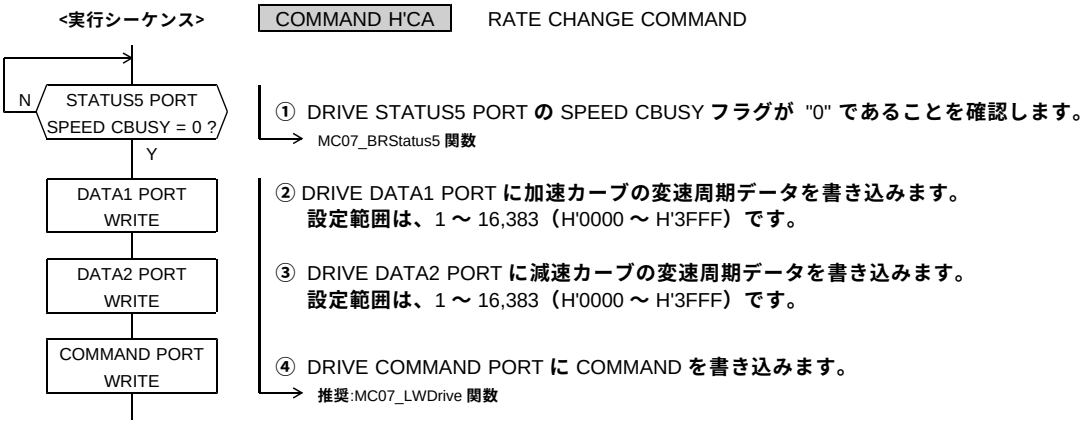
SPEED CHANGE コマンドを実行しても、予め設定されている速度パラメータの値は変わりません。

3-1-9. RATE CHANGE の設定と実行

RATE CHANGE 指令は、スピード系のドライブ CHANGE 指令の検出と同時に実行します。

(1) RATE CHANGE

ドライブ CHANGE 動作時の変更周期データを、指定したデータに変更します。



DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	D13 ←					UCYCLE							→	D0

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	D13 ←					DCYCLE							→	D0

RATE CHANGE データの設定値が "0" の場合は、"1" に補正します。
・加速カーブの変速周期（μs）＝ UCYCLE x 0.5 μs ：0.5 μs ～ 8.1915 ms
・減速カーブの変速周期（μs）＝ DCYCLE x 0.5 μs ：0.5 μs ～ 8.1915 ms

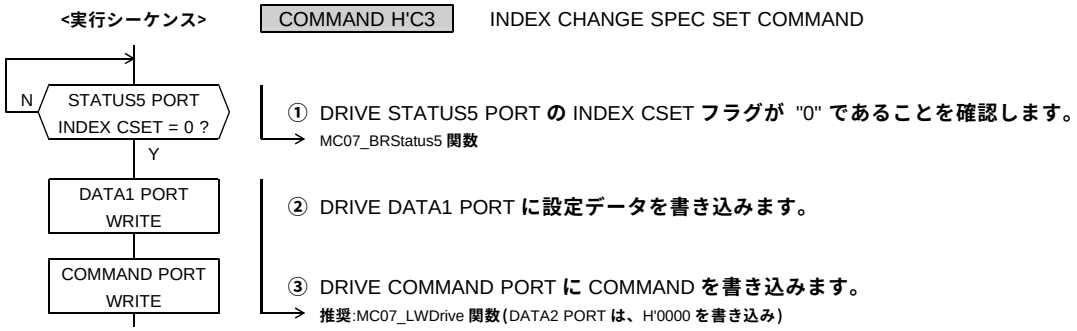
RATE CHANGE コマンドを実行しても、予め設定されている速度パラメータの値は変わりません。

3-1-10. INDEX CHANGE の設定と実行

変更動作点を設定して、INDEX CHANGE を実行します。
変更動作点の設定は、変更動作点の変更が必要な場合に設定します。

(1) INDEX CHANGE SPEC SET

INDEX CHANGE 指令を実行する変更動作点を設定します。



DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	–	–	–	–

D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	INDEX CHANGE TYPE2	INDEX CHANGE TYPE1	INDEX CHANGE TYPE0

●電源投入後の初期値は H'0 (アンダーライン側) です。

D2--D0 ：INDEX CHANGE TYPE2--0
INDEX CHANGE 指令を実行する変更動作点を選択します。

TYPE2	TYPE1	TYPE0	INDEX CHANGE を実行する変更動作点 <レベル検出>
0	0	0	INDEX CHANGE コマンドの書き込みで実行する
0	0	1	設定禁止
0	1	0	STATUS5 PORT の SS0 = 1 で実行する (注)
0	1	1	設定禁止
1	0	0	設定禁止
1	0	1	設定禁止
1	1	0	設定禁止
1	1	1	設定禁止

(注) SS0 信号は SPEC INITIALIZE2 コマンドで「各種トリガ」に設定している場合に有効です。

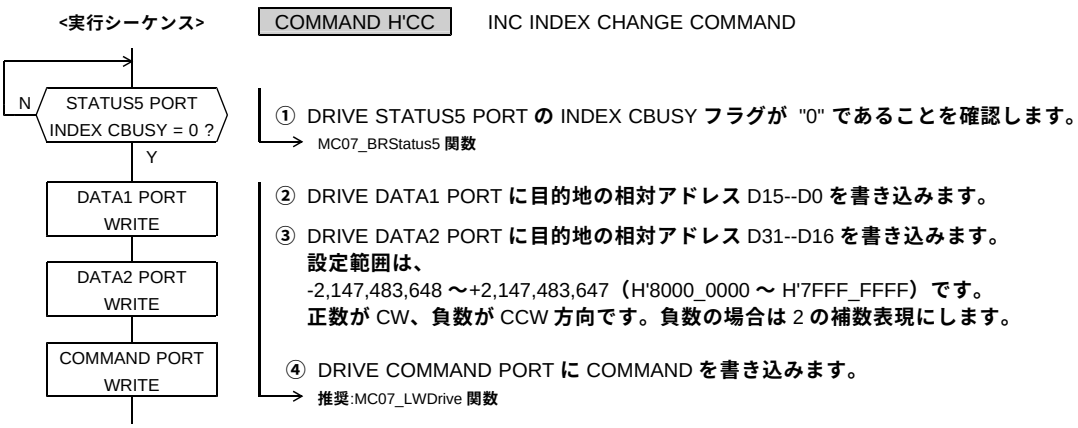
・SS0 信号は下記の軸に対応しています。Z 軸および A 軸には対応していません。

/IN0 信号	X 軸の SS0 信号
/IN1 信号	Y 軸の SS0 信号

D15--D3 ：0
反転方向の INDEX CHANGE 指令は実行できません。

(2) INC INDEX CHANGE

変更動作点の検出で、INC INDEX CHANGE 指令を実行します。
指定したデータを、起動位置を原点とする相対アドレスの停止位置に設定して、
INC INDEX ドライブを行います。



DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15 ←						INDEX CHANGE データ									→ D0

DRIVE DATA2 PORT の設定データ

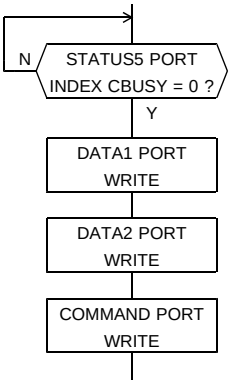
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31 ←						INDEX CHANGE データ									→ D16

指定する相対アドレスは、起動位置から停止位置までのパルス数を、
起動位置を原点として符号付きで表現した値です。

(3) ABS INDEX CHANGE

変更動作点の検出で、ABS INDEX CHANGE 指令を実行します。
指定したデータを、アドレスカウンタで管理している絶対アドレスの停止位置に設定して、
ABS INDEX ドライブを行います。

<実行シーケンス>



ABS INDEX CHANGE COMMAND

① DRIVE STATUS5 PORT の INDEX CBUSY フラグが "0" であることを確認します。
→ MC07_BRStatus5 関数

② DRIVE DATA1 PORT に目的地の絶対アドレス A15--A0 を書き込みます。

③ DRIVE DATA2 PORT に目的地の絶対アドレス A31--A16 を書き込みます。
設定範囲は、
-2,147,483,648 ～+2,147,483,647（H'8000_0000 ～ H'7FFF_FFFF）です。
負数の場合は 2 の補数表現にします。

④ DRIVE COMMAND PORT に COMMAND を書き込みます。
→ 推奨:MC07_LWDrive 関数

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A15	←					INDEX CHANGE データ								→	A0

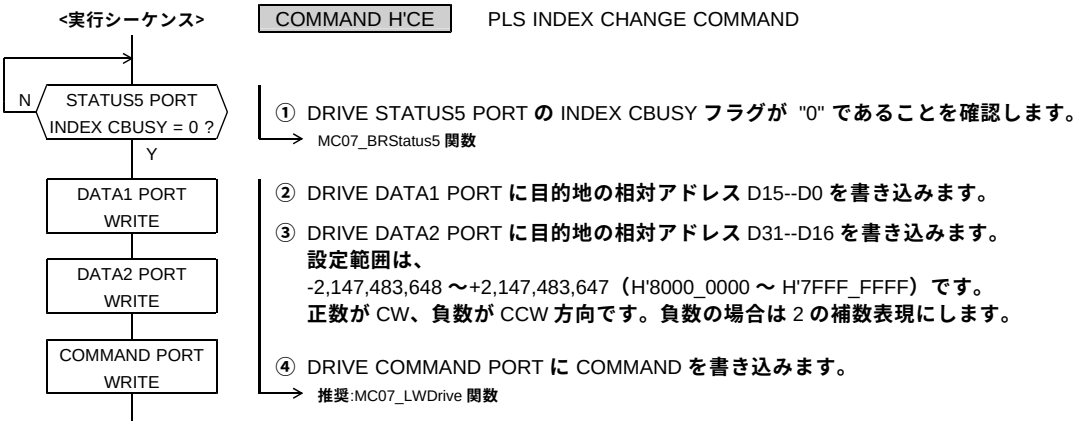
DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A31	←					INDEX CHANGE データ								→	A16

指定する絶対アドレスは、アドレスカウンタで管理している絶対アドレスです。

(4) PLS INDEX CHANGE

変更動作点の検出で、PLS INDEX CHANGE 指令を実行します。
指定したデータを、変更動作点の検出位置を原点とする相対アドレスの停止位置に設定して、
INC INDEX ドライブを行います。



DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15 ←						INDEX CHANGE データ									→ D0

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31 ←						INDEX CHANGE データ									→ D16

指定する相対アドレスは、変更動作点の検出位置から停止位置までのパルス数を、
変更動作点の検出位置を原点として符号付きで表現した値です。

3-2. カウンタコマンド

3-2-1. アドレスカウンタの設定

(1) ADDRESS COUNTER MAX COUNT SET

アドレスカウンタの最大カウント数を設定します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。

<実行シーケンス>

COMMAND H'87　ADDRESS COUNTER MAX COUNT SET COMMAND

DATA1 PORT WRITE

DATA2 PORT WRITE

COMMAND PORT WRITE

① DRIVE DATA1 PORT に最大カウント数 A15--A0 を書き込みます。

② DRIVE DATA2 PORT に最大カウント数 A31--A16 を書き込みます。
設定範囲は、1 ～ 4,294,967,295（H'0000_0001 ～ H'FFFF_FFFF）です。

③ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A15	←						最大カウント数							→	A0

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A31	←						最大カウント数							→	A16

●電源投入後の初期値は H'FFFF_FFFF です。

・カウント数が設定値の 1/2 に達すると、DRIVE STATUS4 PORT の ADDRESS OVF = 1 になります。

・最大カウント数を設定しても、現在のアドレスカウンタの値は変わりません。
アドレスカウンタの値が、最大カウント数の範囲内になったときから、設定が有効になります。

■最大カウント数

設定値をカウンタの最大値として、リングカウントします。
DRIVE STATUS4 PORT の ADDRESS OVF フラグを無視すれば、回転系のアドレス管理ができます。

・最大カウント数＝ 1,999 の場合（2,000 カウントで１回転）
＋方向のカウント：0 → 1 →…→ 999 → 1000 (ADDRESS OVF = 1)→ 1001 →…→ 1999 → 0
－方向のカウント：0 → 1999 →…→ 1001 → 1000 (ADDRESS OVF = 1)→ 999 →…→ 1 → 0

・最大カウント数＝ 2,000 の場合（2,001 カウントで１回転）
＋方向のカウント：0 → 1 →…→ 1000 → 1001 (1001 になると ADDRESS OVF = 1)→…→ 2000 → 0
－方向のカウント：0 → 2000 →…→ 1001 → 1000 (1000 になると ADDRESS OVF = 1)→…→ 1 → 0

3-2-2. パルスカウンタの設定

(1) PULSE COUNTER MAX COUNT SET

パルスカウンタの最大カウント数を設定します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。

<実行シーケンス>

COMMAND H'97　PULSE COUNTER MAX COUNT SET COMMAND

DATA1 PORT WRITE
DATA2 PORT WRITE
COMMAND PORT WRITE

① DRIVE DATA1 PORT に最大カウント数 D15--D0 を書き込みます。

② DRIVE DATA2 PORT に最大カウント数 D31--D16 を書き込みます。
設定範囲は、1 ～ 4,294,967,295（H'0000_0001 ～ H'FFFF_FFFF）です。

③ DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15 ←							最大カウント数							→	D0

DRIVE DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31 ←							最大カウント数							→	D16

●電源投入後の初期値は H'FFFF_FFFF です。

・カウント数が設定値の 1/2 に達すると、DRIVE STATUS4 PORT の PULSE OVF = 1 になります。

・最大カウント数を設定しても、現在のパルスカウンタの値は変わりません。
パルスカウンタの値が、最大カウント数の範囲内になったときから、設定が有効になります。

■最大カウント数

設定値をカウンタの最大値として、リングカウントします。
DRIVE STATUS4 PORT の PULSE OVF フラグを無視すれば、回転系のアドレス管理ができます。

・最大カウント数＝ 1,999 の場合（2,000 カウントで１回転）
＋方向のカウント：0 → 1 →…→ 999 → 1000 (ADDRESS OVF = 1)→ 1001 →…→ 1999 → 0
－方向のカウント：0 → 1999 →…→ 1001 → 1000 (ADDRESS OVF = 1)→ 999 →…→ 1 → 0

・最大カウント数＝ 2,000 の場合（2,001 カウントで１回転）
＋方向のカウント：0 → 1 →…→ 1000 → 1001 (1001 になると ADDRESS OVF = 1)→…→ 2000 → 0
－方向のカウント：0 → 2000 →…→ 1001 → 1000 (1000 になると ADDRESS OVF = 1)→…→ 1 → 0

3-2-3. カウンタのラッチ・クリア機能の設定

設定したラッチタイミングのアクティブエッジで、カウンタのカウントデータをラッチします。
ラッチしたデータは、次のラッチタイミングのアクティブエッジが入力するまで保存します。
ラッチデータは、DRIVE DATA1, 2 PORT（READ）から読み出します。

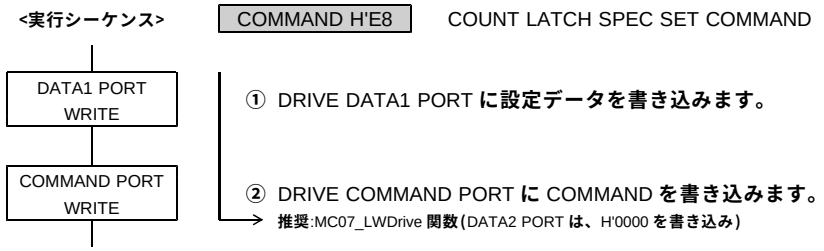
各カウンタには、ラッチタイミングによるカウンタのクリア機能があります。

■カウンタのクリア機能

カウントデータのラッチと同時に、カウンタのデータを "0" にクリアします。
カウンタのカウントとクリアのタイミングが同時に発生した場合は、クリアを優先します。

(1) COUNT LATCH SPEC SET

各種カウンタのカウントデータをラッチするタイミングとクリア機能を設定します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。


<実行シーケンス>
COMMAND H'E8
COUNT LATCH SPEC SET COMMAND
DATA1 PORT WRITE
COMMAND PORT WRITE
① DRIVE DATA1 PORT に設定データを書き込みます。
② DRIVE COMMAND PORT に COMMAND を書き込みます。
推奨:MC07_LWDrive 関数(DATA2 PORT は、H'0000 を書き込み)

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	–	DFL CLR ENABLE	DFL LATCH TYPE2	DFL LATCH TYPE1	DFL LATCH TYPE0

D7	D6	D5	D4	D3	D2	D1	D0
PULSE CLR ENABLE	PULSE LATCH TYPE2	PULSE LATCH TYPE1	PULSE LATCH TYPE0	ADDRESS CLR ENABLE	ADDRESS LATCH TYPE2	ADDRESS LATCH TYPE1	ADDRESS LATCH TYPE0

●電源投入後の初期値は H'00（アンダーライン側）です。

D0 ：ADDRESS LATCH TYPE0
D1 ：ADDRESS LATCH TYPE1
D2 ：ADDRESS LATCH TYPE2

アドレスカウンタのカウントデータをラッチするタイミングを選択します。

TYPE2	TYPE1	TYPE0	ラッチタイミング <エッジ検出>
0	0	0	ADDRESS LATCH DATA READ コマンドの実行でラッチする
0	0	1	設定禁止
0	1	0	STATUS5 PORT の SS0 = 0 → 1 でラッチする
0	1	1	ORIGIN SPEC SET コマンドの ORG 検出信号の検出エッジでラッチする
1	0	0	設定禁止
1	0	1	設定禁止
1	1	0	設定禁止
1	1	1	設定禁止

(注) SS0 信号は SPEC INITIALIZE2 コマンドで「各種トリガ」に設定している場合に有効です。

・SS0 信号は下記の軸に対応しています。Z 軸および A 軸には対応していません。

/IN0 信号	X 軸の SS0 信号
/IN1 信号	Y 軸の SS0 信号

D3 ：ADDRESS CLR ENABLE
カウンタのクリア機能で、アドレスカウンタを「クリアする／クリアしない」を選択します。
0 **：クリアしない**
1 **：クリアする**

D4 ：PULSE LATCH TYPE0
D5 ：PULSE LATCH TYPE1
D6 ：PULSE LATCH TYPE2
パルスカウンタのカウントデータをラッチするタイミングを選択します。

TYPE2	TYPE1	TYPE0	ラッチタイミング <エッジ検出>
0	0	0	PULSE LATCH DATA READ コマンドの実行でラッチする
0	0	1	設定禁止
0	1	0	STATUS5 PORT の SS0 = 0 → 1 でラッチする
0	1	1	ORIGIN SPEC SET コマンドの ORG 検出信号の検出エッジでラッチする
1	0	0	設定禁止
1	0	1	設定禁止
1	1	0	設定禁止
1	1	1	設定禁止

(注) SS0 **信号は** SPEC INITIALIZE2 **コマンドで「各種トリガ」に設定している場合に有効です。**

・SS0 **信号は下記の軸に対応しています。**Z **軸および** A **軸には対応していません。**

/IN0 **信号**	X **軸の** SS0 **信号**
/IN1 **信号**	Y **軸の** SS0 **信号**

D7 ：PULSE CLR ENABLE
カウンタのクリア機能で、パルスカウンタを「クリアする／クリアしない」を選択します。
0 **：クリアしない**
1 **：クリアする**

D8 ：DFL LATCH TYPE0
D9 ：DFL LATCH TYPE1
D10 ：DFL LATCH TYPE2
パルス偏差カウンタのカウントデータをラッチするタイミングを選択します。

TYPE2	TYPE1	TYPE0	ラッチタイミング <エッジ検出>
0	0	0	DFL LATCH DATA READ コマンドの実行でラッチする
0	0	1	設定禁止
0	1	0	STATUS5 PORT の SS0 = 0 → 1 でラッチする
0	1	1	ORIGIN SPEC SET コマンドの ORG 検出信号の検出エッジでラッチする
1	0	0	設定禁止
1	0	1	設定禁止
1	1	0	設定禁止
1	1	1	設定禁止

(注) SS0 **信号は** SPEC INITIALIZE2 **コマンドで「各種トリガ」に設定している場合に有効です。**

・SS0 **信号は下記の軸に対応しています。**Z **軸および** A **軸には対応していません。**

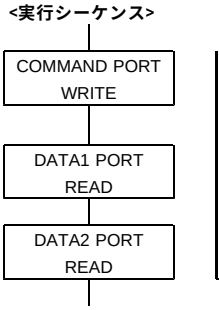
/IN0 **信号**	X **軸の** SS0 **信号**
/IN1 **信号**	Y **軸の** SS0 **信号**

D11 ：DFL CLR ENABLE
カウンタのクリア機能で、パルス偏差カウンタを「クリアする／クリアしない」を選択します。
0 **：クリアしない**
1 **：クリアする**

3-2-4. カウントデータのラッチデータの読み出し

■ラッチデータの読み出しシーケンス

<実行シーケンス>



① DRIVE COMMAND PORT に COMMAND を書き込みます。
(DATA1,2 PORT は、H'0000 にして COMMAND を書き込み)

② DRIVE DATA1 PORT からカウントデータ D15--D0 を読み出します。

③ DRIVE DATA2 PORT からカウントデータ D31--D16 を読み出します。
推奨:MC07_LWRDrive 関数

DRIVE DATA1 PORT の読み出しデータ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						ラッチデータ							→	D0

DRIVE DATA2 PORT の読み出しデータ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←						ラッチデータ							→	D16

・ADDRESS LATCH DATA READ コマンドまたは PULSE LATCH DATA READ コマンドを実行すると、カウンタのラッチデータを DRIVE DATA1, 2 PORT（READ）にセットします。

・DFL LATCH DATA READ コマンドを実行すると、カウンタのラッチデータを DRIVE DATA1 PORT（READ）にセットします。

(1) ADDRESS LATCH DATA READ

アドレスカウンタのラッチデータを読み出します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。

COMMAND H'DC　ADDRESS LATCH DATA READ COMMAND

(2) PULSE LATCH DATA READ

パルスカウンタのラッチデータを読み出します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。

COMMAND H'DD　PULSE LATCH DATA READ COMMAND

(3) DFL LATCH DATA READ

パルス偏差カウンタのラッチデータを読み出します。
このコマンドの実行は DRIVE STATUS1 PORT の BUSY=1 のときも可能です。

COMMAND H'DE　DFL LATCH DATA READ COMMAND

４．機能説明

4-1. ドライブ仕様

4-1-1. コマンド予約機能

MCC07 の各軸には、10 命令分のデータ・コマンドを格納する予約レジスタがあります。
予約レジスタには DRIVE COMMAND の汎用コマンドを予約することができます。
＊ DRIVE COMMAND の特殊コマンドは予約できません。

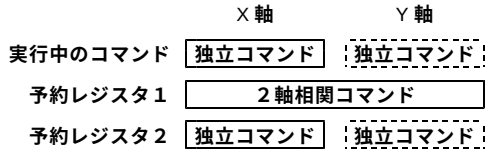
予約レジスタの状態は、DRIVE STATUS1 PORT の COMREG EP と COMREG FL フラグで確認します。

BUSY = 1 で、COMREG FL = 0 のときに、DRIVE COMMAND PORT に汎用コマンドを書き込むと、
DRIVE DATA1, 2 PORT のデータと汎用コマンドの１命令分を、予約レジスタに格納します。

予約レジスタは FIFO 構成になっています。
実行中のコマンド処理が終了すると、予約レジスタに格納したコマンドを順次実行します。

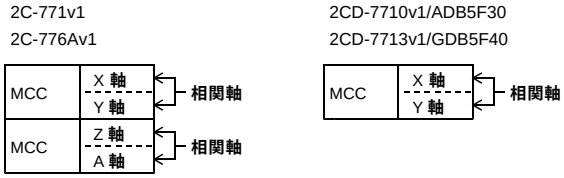
● 2 軸相関コマンドの予約
2 軸相関コマンドは、相関軸 2 軸の COMREG FL = 0 のときに予約することができます。
2 軸相関コマンドの予約は、以下のように実行します。

・2 軸相関コマンドの命令は、相関軸両軸の予約レジスタに格納します。
・片方の軸の次に実行する予約コマンドが 2 軸相関コマンドになっても、もう片方の軸が別のコマンド処理中の場合は、BUSY = 1 のまま、もう片方の軸のコマンド処理終了を待ちます。
・両軸の次に実行する予約コマンドが 2 軸相関コマンドになると、2 軸相関コマンドを実行します。



以下の 2 軸相関コマンド実行中は、相関軸のどちらの軸で停止指令が発生しても有効になります。
・MAIN XY STRAIGHT CP コマンド
・MAIN XY CIRCULAR CP コマンド

当製品での相関軸は以下の通りです。



●コマンド予約機能が自動的に無効となる状態
DRIVE STATUS1 PORT の ERROR = 1 になると、予約コマンドをすべてクリアします。
また、実行待ちの予約コマンドをクリアした場合は、ERROR STATUS の COMREG CLR ERROR = 1 にします。

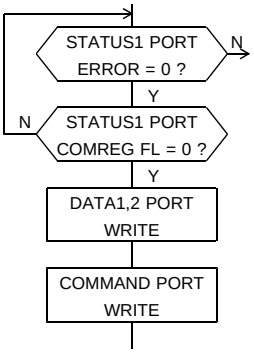
2 軸相関コマンドを予約している場合は、ERROR = 1 で X, Y 軸(X,A 軸)の予約コマンドをすべてクリアします。
エラーが発生していない軸も、COMREG CLR ERROR = 1 により ERROR = 1 になります。

ERROR = 1 の間は、COMREG FL = 1、COMREG EP = 1 になり、予約コマンド（汎用コマンド）の書き込みが無効になります。
ERROR = 0 にクリアすると、COMREG FL = 0 になり、予約コマンドの書き込みが有効になります。

■ コマンド予約機能の実行例

汎用コマンドを予約する場合は、BUSY = 0 の代わりに COMREG FL = 0 を確認します。
予約シーケンス実行中に BUSY = 0 になった場合は、通常のコマンド実行と同様になります。

●コマンド予約シーケンス



① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。

② DRIVE STATUS1 PORT の COMREG FL フラグが "0" であることを確認します。

③ COMMAND に必要なデータを DRIVE DATA1,2 PORT に書き込みます。

④ DRIVE COMMAND PORT に COMMAND を書き込みます。

● INDEX ドライブ → JOG ドライブの連続実行例

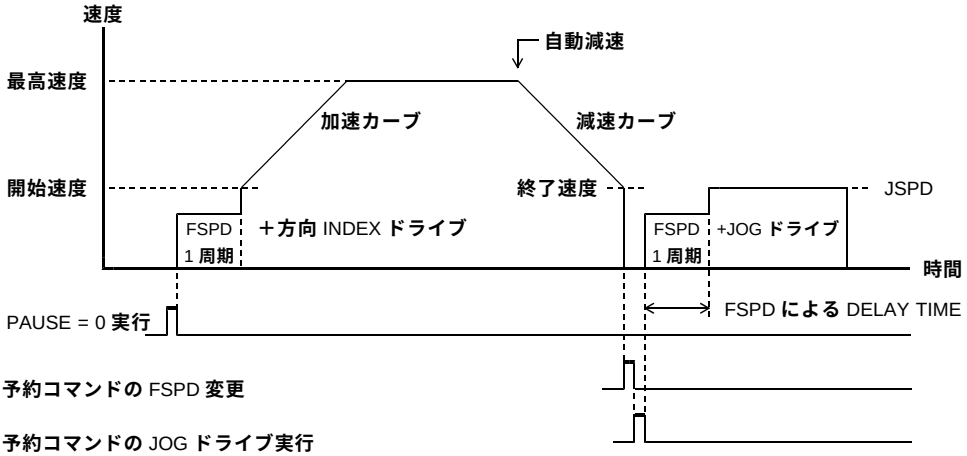
① HARD INITIALIZE7 コマンドのアクティブ論理変更で対象軸の PAUSE を 1 にします。

② ＋方向 INDEX ドライブを実行します。
STBY = 1 のままドライブが保留になります。

③ FSPD による DELAY TIME を設定します。

④ ＋方向 JOG ドライブを実行します。

⑤ HARD INITIALIZE7 コマンドのアクティブ論理変更で対象軸の PAUSE を 0 にします。
STBY = 0 になり、INDEX ドライブ → JOG ドライブを開始します。



4-1-2. 入出力仕様

(1) 入力信号のデジタルフィルタ機能

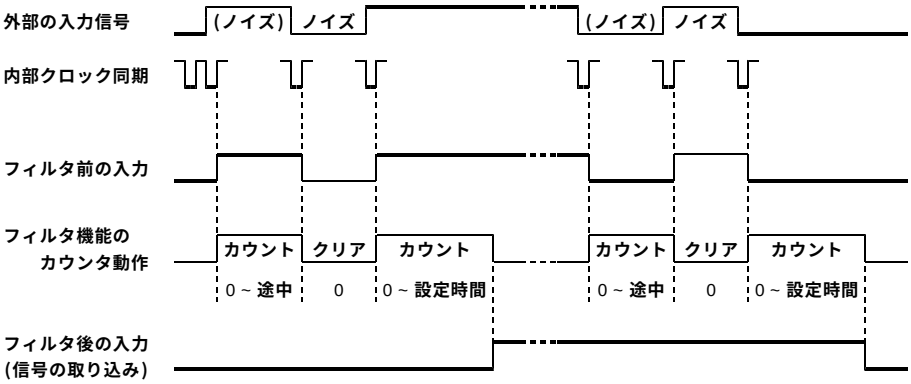
AL-Ⅱシリーズのコントローラ入力回路には、ハード的なノイズフィルタが入っており、原則設定不要です。
環境下によって不要な信号を拾うようなとき、デジタルフィルタを設定することができます。
デジタルフィルタを設定すると設定時定数により、信号の応答性は低下します。

以下の入力信号に対してデジタルフィルタを設定できます。

入力信号	設定可能範囲	対応コマンド
CWLM, CCWLM 信号 $\overline{\text{DEND}}$/$\overline{\text{PO}}$, DALM 信号	0 ～ 10ms	HARD INITIALIZE4 コマンド
$\overline{\text{ORG}}$,$\overline{\text{NORG}}$ 信号	0 ～ 10ms	HARD INITIALIZE5 コマンド
± ZORG 信号	0 ～ 1ms	HARD INITIALIZE5 コマンド
± EA,± EB 信号	0 ～ 12.75 μs	HARD INITIALIZE6 コマンド

●電源投入後の初期値はアンダーライン側です。

■デジタルフィルタ機能



・入力信号が L → H、または H → L に変化すると、フィルタ機能のカウントを開始して
入力信号のレベルを計測します。
フィルタ機能の設定時間分のカウントが終了すると、入力信号のレベルを取り込みます。

・計測の途中で、レベルが変化（ノイズが入力）すると、フィルタ機能のカウンタを
クリアして計測を中止します。この場合は、入力信号のレベルを取り込みません。

(2) 入力信号の論理切り替え機能

下記の入力信号のアクティブ論理を初期値から切り替えることができます。

入力信号	初期値	対応コマンド
FSSTOP 信号	正論理入力	HARD INITIALIZE7 コマンド
CWLM 信号	正論理入力	
CCWLM 信号	正論理入力	
DALM 信号	正論理入力	
ORG 信号	負論理入力	
NORG 信号	負論理入力	
PAUSE 信号	1 でアクティブ	

・PAUSE 信号は外部とインターフェースする信号ではありません。
コマンド予約機能で操作する MCC07 の入力信号です。
・アクティブ論理を変更すると、変更した信号のデジタルフィルタ機能が動作します。
デジタルフィルタ機能の時定数経過後に、アクティブ論理の変更が確定します。

4-1-3. ドライブパラメータ
(1) 加減速パラメータ

加減速ドライブは、加速カーブと減速カーブで加減速を行うドライブです。
・加速カーブは、S字加速の変速領域と直線加速の変速領域で構成します。
・減速カーブは、S字減速の変速領域と直線減速の変速領域で構成します。
・加速カーブと減速カーブを異なる設定にすると、非対称の加減速ドライブになります。

加減速ドライブには、以下のドライブパラメータの設定が必要です。

● 速度のパラメータ
・FSPD ：第１パルスの速度（Hz）

・RESOL ：速度データの速度倍率
・HSPD ：最高速時のパルス速度データ
・LSPD ：加速開始時のパルス速度データ
・ELSPD ：減速終了時のパルス速度データ

・JSPD ：JOG ドライブの速度（Hz）

● 加減速カーブのパラメータ
・UCYCLE ：加速カーブの変速周期データ
・DCYCLE ：減速カーブの変速周期データ

・SUAREA ：加速カーブのS字変速領域データ
・SDAREA ：減速カーブのS字変速領域データ

● 減速パルス数の調整／設定パラメータ
・DOWN PULSE ADJUST ：自動減速パルス数のオフセットパルス数

■ 加減速ドライブの速度とS字領域

加減速ドライブのパルス速度は、速度データと速度倍率で設定します。
・最高速時の速度 （Hz）＝ HSPD x RESOL
・加速開始時の速度（Hz）＝ LSPDx RESOL
・減速終了時の速度（Hz）＝ ELSPD x RESOL

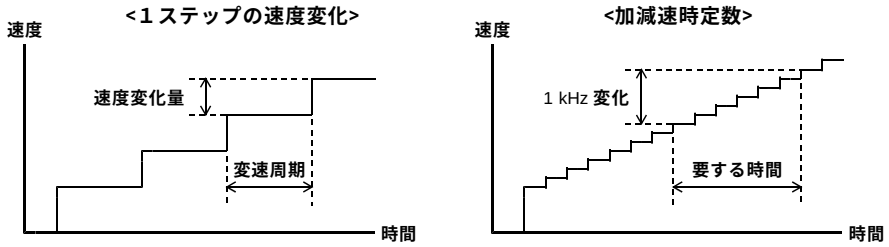
・SPEED CHANGE の指定速度（Hz）＝ SPEED CHANGE データ x RESOL

S字領域は、S字変速領域データと速度倍率で設定します。
・S字加速開始部の変速領域（Hz）＝ SUAREA x RESOL
・S字加速終了部の変速領域（Hz）＝ SUAREA x RESOL

・S字減速開始部の変速領域（Hz）＝ SDAREA x RESOL
・S字減速終了部の変速領域（Hz）＝ SDAREA x RESOL

(2) 加減速時定数

加速および減速は、速度変化量を変速周期毎に加算および減算することで行っています。
加減速時定数は、速度を 1 kHz 変化させるのに要する時間（ms/kHz）で表しています。
本資料では、この時定数を RATE と呼称しています。



● 加減速時定数の設定方法

①最高速度と速度倍率を設定します。設定した速度倍率データで、速度変化量が決定します。
速度倍率を小さくすると、加減速が滑らかになります。

②開始速度と終了速度を設定します。

③加速の変速周期と減速の変速周期を設定します。設定した変速周期で、加減速時間が決定します。
速度変化量と変速周期で、目的に合った加減速時定数（加減速時間）を設定します。

■ 加減速 RATE の計算式

速度倍率データ（RESOL）と変速周期データ（UCYCLE）で、任意の加速 RATE を設定します。
速度倍率データ（RESOL）と変速周期データ（DCYCLE）で、任意の減速 RATE を設定します。

● 加速カーブの直線加速 RATE の計算式

・直線加速 RATE（ms/kHz）＝ $\frac{\text{加速カーブの変速周期（ms）}}{\text{直線加速カーブの速度変化量（kHz）}} = \frac{\text{UCYCLE}}{\text{RESOL} * 2}$

加速カーブの変速周期 （ms） ＝ UCYCLE * 0.5 * 10^-3

直線加速カーブの速度変化量（kHz）＝ RESOL * 10^-3

● 減速カーブの直線減速 RATE の計算式

・直線減速 RATE（ms/kHz）＝ $\frac{\text{減速カーブの変速周期（ms）}}{\text{直線減速カーブの速度変化量（kHz）}} = \frac{\text{DCYCLE}}{\text{RESOL} * 2}$

減速カーブの変速周期 （ms） ＝ DCYCLE * 0.5 * 10^-3

直線減速カーブの速度変化量（kHz）＝ RESOL * 10^-3

■ RATE DATA TABLE(参考)

RATE (ms/kHz)	RESOL = 1 U/D CYCLE	RESOL = 5 U/D CYCLE	RESOL = 10 U/D CYCLE	RESOL = 20 U/D CYCLE	RESOL = 50 U/D CYCLE	RESOL = 200 U/D CYCLE
5,000	10,000					
3,000	6,000					
2,000	4,000					
1,000	2,000	10,000				
500	1,000	5,000	10,000			
300	600	3,000	6,000	12,000		
200	400	2,000	4,000	8,000		
100	200	1,000	2,000	4,000	10,000	
50	100	500	1,000	2,000	5,000	
30	60	300	600	1,200	3,000	12,000
20	40	200	400	800	2,000	8,000
10	20	100	200	400	1,000	4,000
5	10	50	100	200	500	2,000
3	6	30	60	120	300	1,200
2	4	20	40	80	200	800
1	2	10	20	40	100	400
0.5	1	5	10	20	50	200
0.3		3	6	12	30	120
0.2		2	4	8	20	80
0.1		1	2	4	10	40
0.05			1	2	5	20
0.03					3	12
0.02					2	8
0.01					1	4
0.005						2
0.0025						1

(3) 直線加減速ドライブ

直線加減速ドライブは、S字加減速の変速領域を "0" に設定して、加減速を行うドライブです。
・開始速度から最高速度まで、S字変速領域がない直線加速カーブで加速します。
・最高速度から終了速度まで、S字変速領域がない直線減速カーブで減速します。

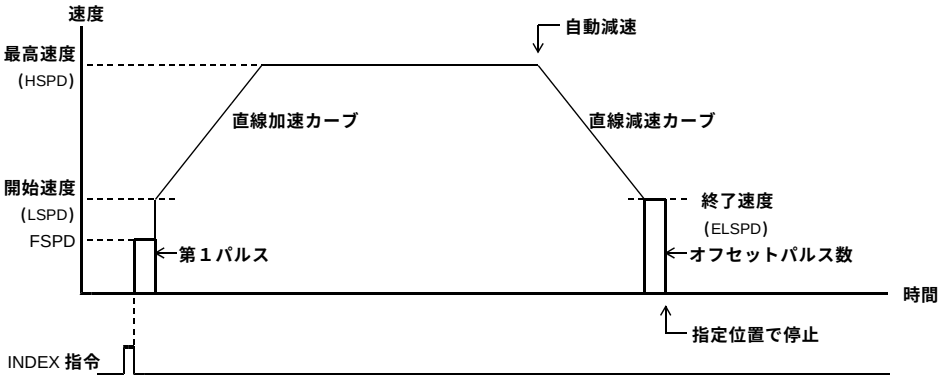
● 直線加速カーブ
SCAREA SET コマンドの SUAREA を "0" に設定します。

開始速度から最高速度まで、UCYCLE の直線加速カーブで加速します。

● 直線減速カーブ
SCAREA SET コマンドの SDAREA を "0" に設定します。

最高速度から終了速度まで、DCYCLE の直線減速カーブで減速します。

■ 直線加減速ドライブの動作



● 直線加減速ドライブの加速時間と減速時間

直線加速カーブの加速時間（ms）：0 ≦ TU ＜ 最高速度の１周期
= (UCYCLE * 0.5 * 10^-3) * (HSPD - LSPD + 1) + TU + 第１パルスの周期(ms)

直線減速カーブの減速時間（ms）：0 ≦ TD ＜ 終了速度の１周期
= (DCYCLE * 0.5 * 10^-3) * (HSPD - ELSPD + 1) + TD

・オフセットパルス数 = 0 で減速停止する場合の減速時間です。
・INDEX ドライブの自動減速停止時には、オフセットパルス数（初期値：+1）の増減があります。

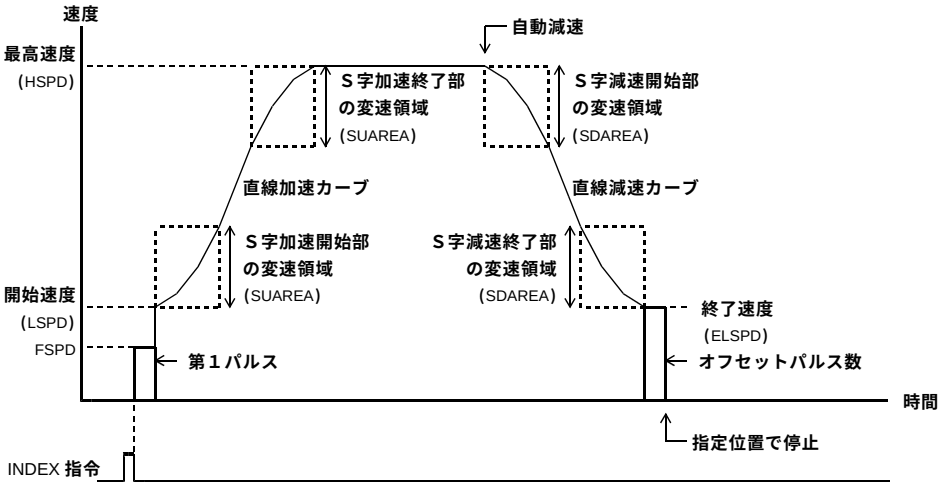
(4) S 字加減速ドライブ

S字加減速ドライブは、S字加減速の変速領域を設定して、加減速を行うドライブです。
・加速開始時の S 字変速領域と加速終了時の S 字変速領域を、放物線に近似した S 字加速カーブで加速し、
・減速開始時の S 字変速領域と減速終了時の S 字変速領域を、放物線に近似した S 字減速カーブで減速します。

● S字加速カーブ
SCAREA SET コマンドの SUAREA で S 字加速の変速領域を設定します。
SUAREA で設定した変速領域が、加速開始時の S 字変速領域と加速終了時の S 字変速領域になり、
S 字加速カーブを形成します。
残りの速度領域は、UCYCLE の直線加速カーブで加速します。

● S字減速カーブ
SCAREA SET コマンドの SDAREA で S 字減速の変速領域を設定します。
SDAREA で設定した変速領域が、減速開始時の S 字変速領域と減速終了時の S 字変速領域になり、
S 字減速カーブを形成します。
残りの速度領域は、DCYCLE の直線減速カーブで減速します。

■ S字加減速ドライブの動作



● S 字加減速ドライブの加速時間と減速時間

S 字加速カーブの加速時間（ms）
＝（UCYCLE x 0.0005） × （HSPD − LSPD ＋ 1 ＋ SUAREA x 2） ＋ 第１パルスの周期(ms)＋ TU
：0 ≦ TU ＜ 最高速度の１周期

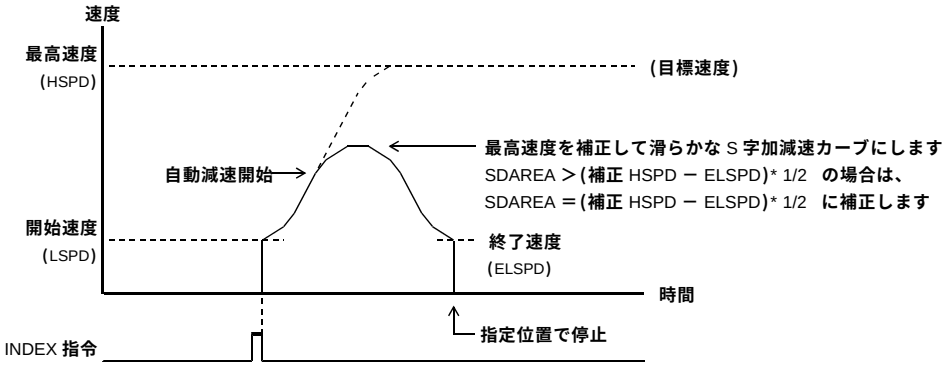
・SUAREA ＜(HSPD − LSPD)／ 2 で、加速する場合の加速時間です。

S 字減速カーブの減速時間（ms）
＝（DCYCLE x 0.0005） × （HSPD − ELSPD ＋ 1 ＋ SDAREA x 2） ＋ TD
：0 ≦ TD ＜ 終了速度の１周期

・SDAREA ＜(HSPD − ELSPD)／ 2 で、減速停止指令で減速停止する場合の減速時間です。
・INDEX ドライブの自動減速停止時には、オフセットパルス数分の増減があります。

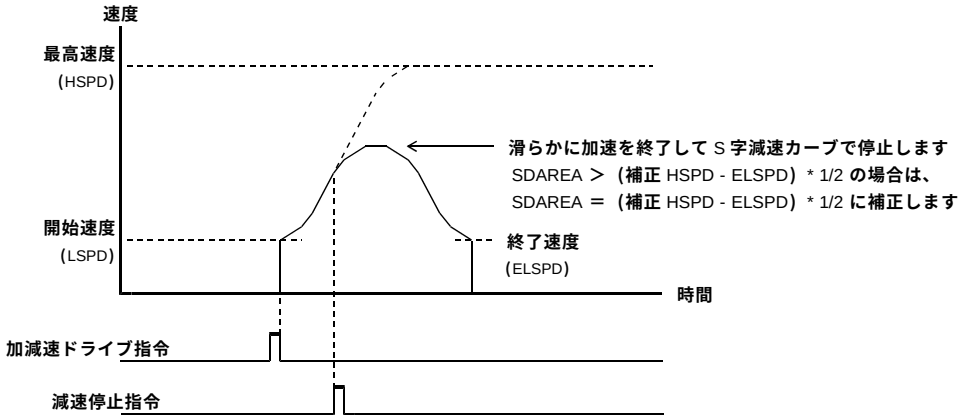
■ S 字加減速 INDEX ドライブの三角駆動回避動作

S 字加減速の INDEX ドライブで、停止位置までのパルス数が少なくて最高速度（目標速度）に達しない場合は自動的に最高速度を引き下げて、滑らかな S 字加減速カーブで INDEX ドライブを停止します。
この機能は常時有効です。



■減速停止指令検出時の三角駆動回避動作

S 字加速中に減速停止指令を検出した場合は、SUAREA の S 字加速終了カーブで滑らかに加速を終了し、
S 字減速カーブで減速停止します。
この機能は常時有効です。

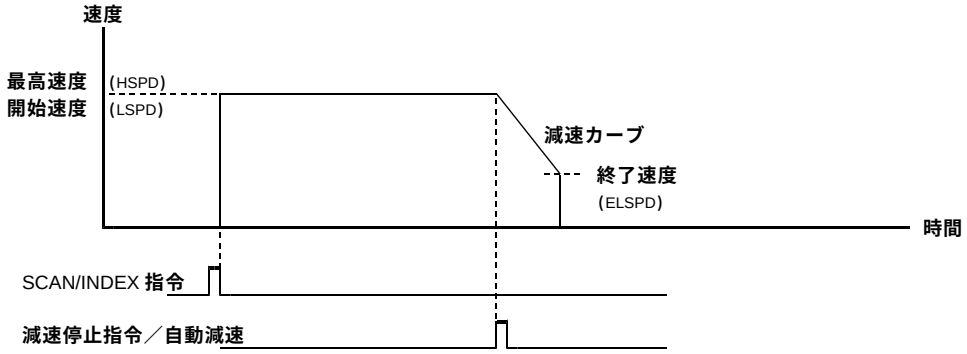


(5) 加速ドライブ

「開始速度＜最高速度」および「最高速度＝終了速度」に設定すると、
開始速度と最高速度による加速ドライブを行います。

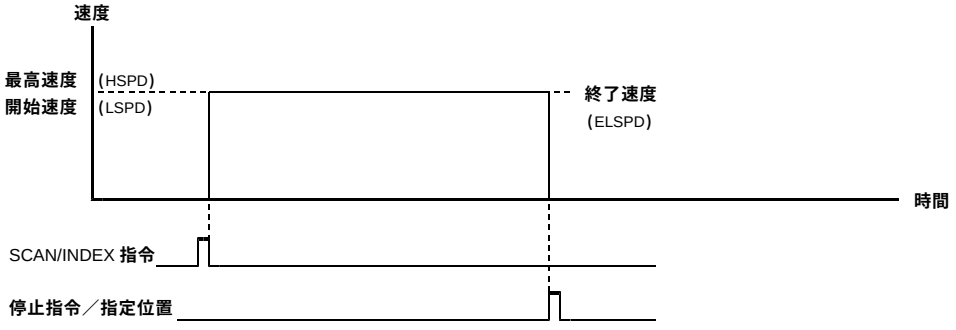
(6) 減速ドライブ

「開始速度＝最高速度」および「最高速度＞終了速度」に設定すると、
最高速度と終了速度による減速ドライブを行います。



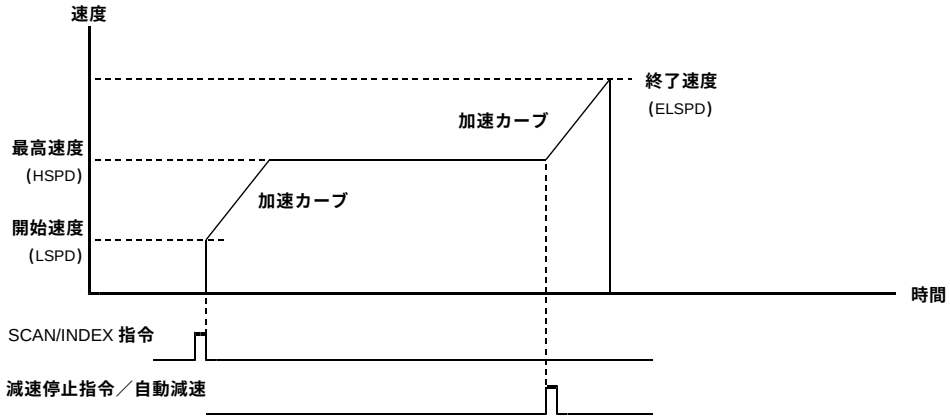
(7) 一定速ドライブ

「開始速度＝最高速度＝終了速度」に設定すると、
開始速度での一定速ドライブを行います。

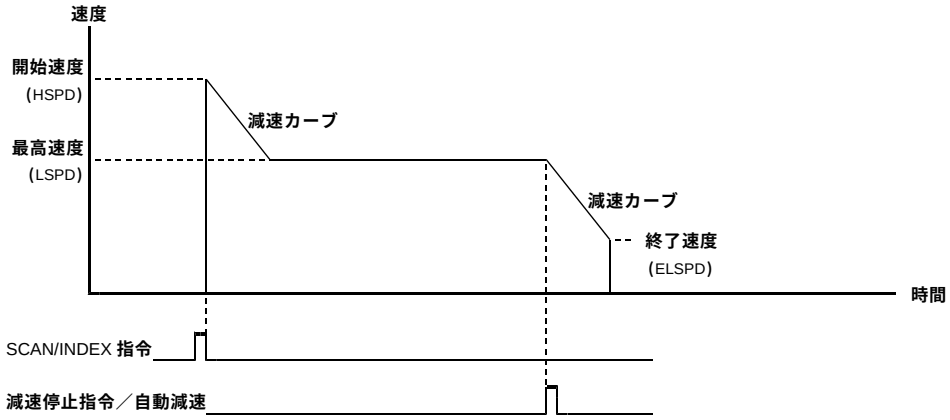


(8) その他のドライブ

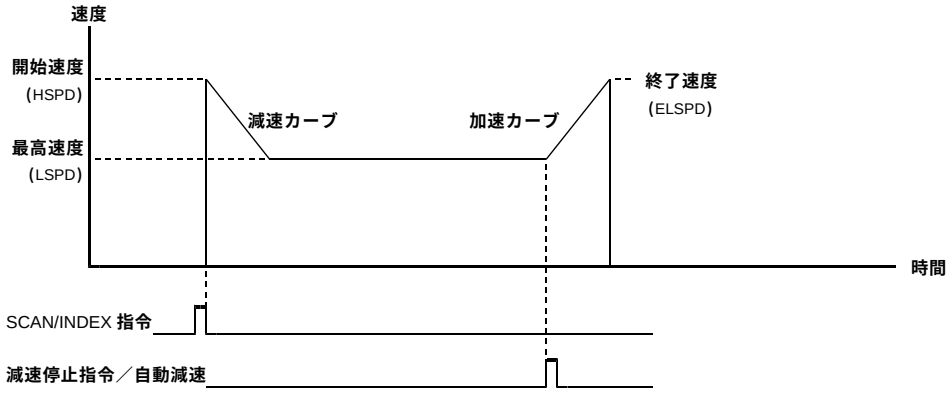
●「開始速度＜最高速度＜終了速度」に設定すると、以下の加速ドライブを行います。



●「開始速度＞最高速度＞終了速度」に設定すると、以下の減速ドライブを行います。



●「開始速度＞最高速度」および「最高速度＜終了速度」に設定すると、
以下の加減速ドライブを行います。



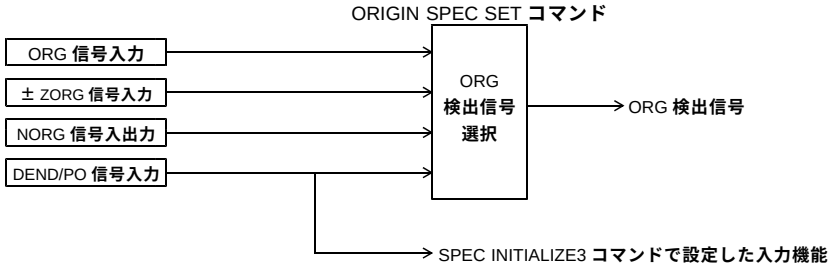
4-1-4. ORIGIN ドライブ　（MCC07 チップの機械原点検出機能）

関数による ORIGIN ドライブ機能は、取扱説明書をご覧ください。

ドライブ工程を指定して ORIGIN ドライブコマンドを実行すると、ORG 検出信号の指定エッジを検出してドライブを終了します。
検出する ORG 検出信号は、$\overline{\text{ORG}}$ 信号, ± ZORG 信号, $\overline{\text{DEND}}$/$\overline{\text{PO}}$ 信号, $\overline{\text{NORG}}$ 信号の合成信号から選択します。

ORIGIN ドライブには、以下のドライブパラメータの設定が必要です。
・加減速ドライブのパラメータ
・JSPD　：JOG ドライブのパルス速度

ORG エッジ信号の検出条件と停止機能は、ORIGIN SPEC SET コマンドで設定します。
・$\overline{\text{DEND}}$/$\overline{\text{PO}}$ 信号を ORG 検出信号に選択した場合は、$\overline{\text{DEND}}$/$\overline{\text{PO}}$ 信号の入力機能と
ORG 検出信号の停止機能の両方が有効になります。

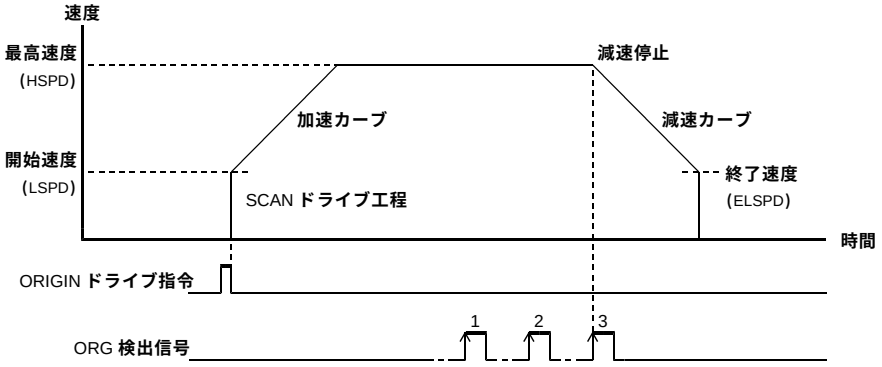


■ ORIGIN 停止機能による原点検出工程

ドライブ工程は、ORIGIN SCAN ドライブと ORIGIN CONSTANT SCAN ドライブが選択できます。

＜ドライブ工程の実行例＞
ORG 検出信号の３カウント目のアクティブエッジ検出で、停止機能を動作させます。
・ORG DETECT EDGE = 0　　：ORG 検出信号の 0 → 1（アクティブ）エッジを検出する
・ORIGIN COUNT D3--D0 = H'2：３カウント目のエッジ検出で、停止機能を動作させる

● ORIGIN SCAN ドライブ
加減速ドライブのパラメータで、SCAN ドライブを行います。
ORG 検出信号の指定エッジを検出すると減速停止します。



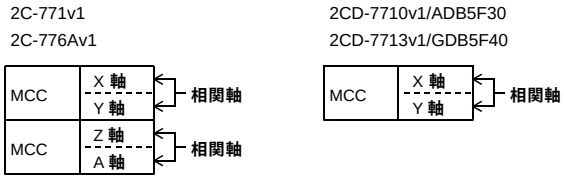
● ORIGIN CONSTANT SCAN ドライブ
JOG ドライブのパルス速度（JSPD）で、一定速ドライブを行います。
ORG 検出信号の指定エッジを検出すると即時停止します。

4-1-5. 補間ドライブ

補間ドライブには、相関軸で実行する相関 2 軸補間ドライブと、任意軸間で実行する任意軸補間ドライブがあります。

(1) 相関 2 軸補間ドライブ

相関軸で行う補間ドライブです。
当製品での相関軸は以下の通りです。



相関軸で 2 軸直線補間ドライブ、および 2 軸円弧補間ドライブが行えます。
　相関 2 軸直線補間ドライブ … MAIN XY STRAIGHT CP コマンド
　相関 2 軸円弧補間ドライブ … MAIN XY CIRCULAR CP コマンド

・相関 2 軸補間ドライブは、コマンド実行軸の加減速パラメータで、補間ドライブの基本パルスを発生します。
補間制御部は、発生した基本パルスを補間演算して補間パルスを出力します。

・相関 2 軸補間ドライブのコマンドは、相関軸のどちらの軸に実行しても有効です。
コマンドの実行で 2 軸のドライブを開始します。

・同期スタート機能は、コマンド実行軸で有効です。他軸の同期スタート機能は無効になります。
コマンド実行軸の STBY = 0 で、他軸も STBY = 0 になります。

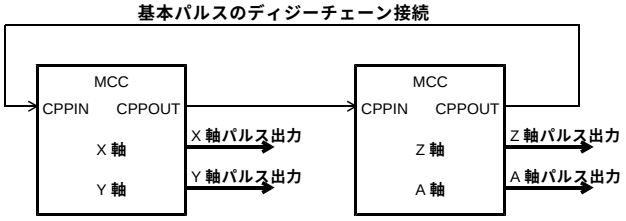
・LIMIT 停止指令、減速停止指令、即時停止指令は、相関軸両軸のどちらで発生しても有効です。

・エラーが発生した場合は、エラー該当軸が ERROR = 1 になります。
但し、エラーによる停止機能は、X, Y 軸のどちらでエラーが発生しても有効です。

・両軸のドライブが終了すると、両軸が BUSY = 0 になります。
$\overline{\text{DEND}}$/$\overline{\text{PO}}$ 信号または $\overline{\text{DRST}}$ 信号を<サーボ対応>に設定している場合は、両軸の<サーボ対応>が終了した後に、両軸が BUSY = 0 になります。

(2) 任意軸補間ドライブ

MCC の CPPOUT 端子から出力するパルスを設定することで任意軸の補間を行うことができます。
4 軸コントローラ製品は、各軸 MCC の CPPOUT 端子と CPPIN 端子はディジーチェーン接続で繋がっています。
任意軸補間ドライブではメイン軸のチップが CPPOUT 端子に補間ドライブの基本パルスを出力します。
サブ軸のチップは基本パルスを CPPIN 端子から入力して CPPOUT 端子に出力します。



任意多軸直線補間ドライブ … MAIN STRAIGHT CP コマンド (メイン軸)
SUB STRAIGHT CP コマンド (サブ軸)
任意 2 軸円弧補間ドライブ … MAIN CIRCULAR CP コマンド (メイン軸)
SUB CIRCULAR CP コマンド (サブ軸)

・任意軸補間ドライブは、メイン軸の加減速パラメータで、補間ドライブの基本パルスを発生します。
補間制御部は、発生した基本パルスを補間演算して補間パルスを出力します。

・サブ軸は、CPPIN 端子から入力するパルスを補間ドライブの基本パルスにします。
補間制御部は、CPPIN 端子の入力パルスを補間演算して補間パルスを出力します。

・メイン軸補間ドライブは、各サブ軸にサブ軸補間ドライブを実行した後に、メイン軸に実行してください。
メイン軸はコマンドの実行で任意軸補間ドライブを開始します。

● CPPOUT 出力
CP SPEC SET コマンドの CPPOUT SEL で選択したパルスを出力します。

● CPPIN 入力
補間ドライブの基本パルスを入力します。
CPPIN 端子に入力できるパルス速度の最高値は、5 MHz です。

■任意軸補間ドライブの停止機能

●メイン軸で停止指令が発生した場合

・減速停止指令を検出した場合は、メイン軸の基本パルスを減速停止して、ドライブを終了します。

・即時停止指令を検出した場合は、メイン軸の基本パルスがハイレベルのときに、パルス出力を停止して、ドライブを終了します。

・停止指令の発生したメイン軸はパルス停止後にドライブを終了しますが、サブ軸はドライブを終了しません。
(但し、基本パルスが停止するのでサブ軸のパルス出力は停止します。)
メイン軸が停止指令により補間ドライブを終了した場合は、サブ軸すべてに停止指令を実行して、ドライブを終了させてください。

●サブ軸で停止指令が発生した場合

・減速停止指令を検出した場合は、出力中の補間パルスがハイレベルのときに、ドライブを終了します。

・即時停止指令を検出した場合は、出力中の補間パルスがハイレベルのときに、ドライブを終了します。

・停止指令の発生したサブ軸はパルス停止後にドライブを終了しますが、他の補間軸はドライブを終了しません。
(メイン軸の基本パルスは停止していないので、メイン軸および他のサブ軸のパルス出力も停止しません。)
サブ軸が停止指令により補間ドライブを終了した場合は、他のすべての補間軸に停止指令を実行して、ドライブを終了させてください。

● CPP STOP 機能と CPPIN マスク機能

メイン軸の CPP STOP 機能とサブ軸の CPPIN マスク機能を有効にして、任意軸補間ドライブを実行すると、サブ軸にエラーが発生した場合に、すべての補間軸のパルス出力を停止させることができます。

・エラーが発生したサブ軸は、CPPIN マスク機能で CPPOUT 出力を停止します。

・メイン軸は、CPP STOP 機能でドライブを終了し、CPPOUT 出力を終了します。

・他のサブ軸は、メイン軸の CPPOUT 出力終了でパルス出力を停止します。

(3) 直線補間ドライブ

補間軸は任意の長軸と短軸の座標を構成し、指定軸のパルスを出力して直線補間します。
補間ドライブの最高速度は、5MHz です。指定直線に対する位置誤差は、± 0.5 LSB です。
座標指定できる相対アドレス範囲は、-2,147,483,648 ～ +2,147,483,647（32 ビット）です。
長軸のパルス出力が INDEX ドライブと同様の加減速ドライブとなります。

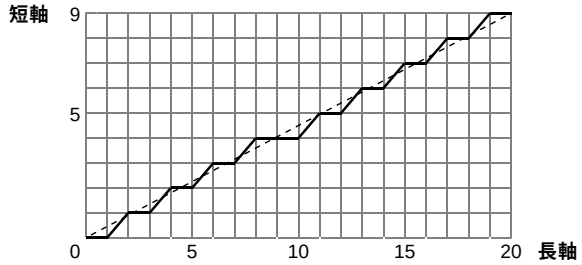
直線補間ドライブの基本パルスは、以下のようになります。
・相関 2 軸直線補間ドライブは、コマンド実行軸から基本パルスを発生します。
　相関 2 軸とも実行軸の基本パルスを補間演算して補間パルスを出力します。
・任意軸補間ドライブのメイン軸直線補間ドライブは、コマンド実行軸から基本パルスを発生します。
・任意軸補間ドライブのサブ軸直線補間ドライブは、CPPIN 端子から入力するパルスを基本パルスとします。

相関 2 軸直線補間ドライブには、以下のドライブパラメータの設定が必要です。
・実行軸の加減速ドライブのパラメータ
・XLONG POSITION、XSHORT POSITION：X 補間軸の長軸と短軸の座標アドレス
・YLONG POSITION、YSHORT POSITION：Y 補間軸の長軸と短軸の座標アドレス

任意軸補間ドライブのメイン軸直線補間ドライブには、以下のドライブパラメータの設定が必要です。
・実行軸の加減速ドライブのパラメータ
・CP SPEC　　：CPPOUT 出力
・LONG POSITION、SHORT POSITION　：補間軸の長軸と短軸の座標アドレス

任意軸補間ドライブのサブ軸直線補間ドライブには、以下のドライブパラメータの設定が必要です。
・CP SPEC　　：CPPOUT 出力
・LONG POSITION、SHORT POSITION　：補間軸の長軸と短軸の座標アドレス

■直線補間ドライブの軌跡(長軸 20:短軸 9 の例)



直線補間ドライブの軌跡は、現在位置と目的地を結ぶ直線に沿います。
・直線補間 SCAN ドライブの場合は、停止指令を検出するまで目的地の指定方向にパルス出力を続けます。
・直線補間 INDEX ドライブの場合は、長軸のパルス数が目的地のパルス数になるとドライブを終了します。

●直線補間の長軸と短軸
補間パルス数が大きい方の軸が長軸、小さい方の軸が短軸になります。

(4) 円弧補間ドライブ

現在の座標と中心点で形成する円弧曲線上を、指定の短軸パルス数に達するまで円弧補間します。
補間ドライブの最高速度は、5MHz です。指定円弧曲線に対する位置誤差は、± 1 LSB です。
座標指定できる相対アドレス範囲は、-8,388,608 ～ +8,388,607（24 ビット）です。
短軸パルス数の設定範囲は、-2,147,483,648 ～ +2,147,483,647（32 ビット）です。
短軸パルス出力が INDEX ドライブと同様の加減速ドライブとなります。

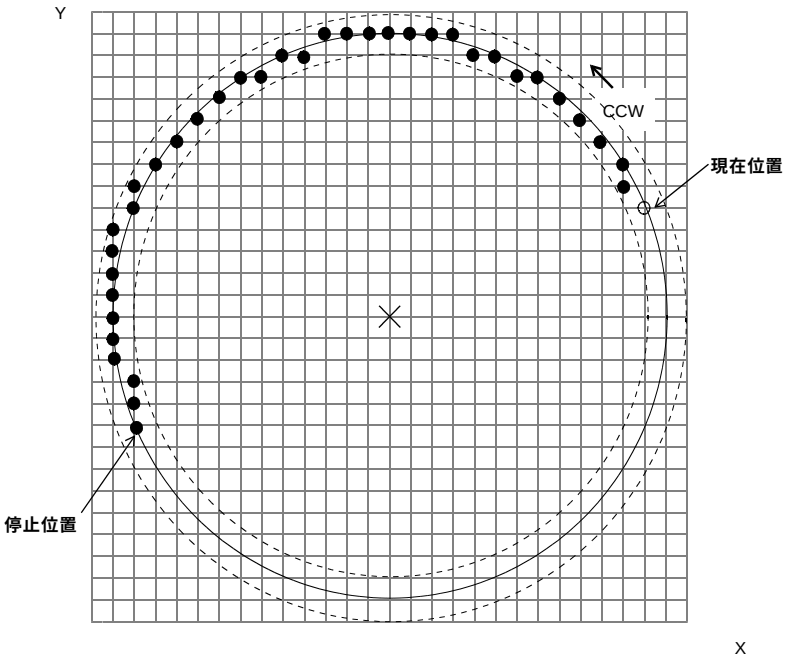
円弧補間ドライブの基本パルスは、以下のようになります。
・相関 2 軸円弧補間ドライブは、コマンド実行軸から基本パルスを発生します。
　相関 2 軸は実行軸の基本パルスを補間演算して XCP, YCP の補間パルスを出力します。
・任意軸補間ドライブのメイン軸円弧補間ドライブは、コマンド実行軸から基本パルスを発生します。
・任意軸補間ドライブのサブ軸円弧補間ドライブは、CPPIN 端子から入力するパルスを基本パルスとします。

相関 2 軸円弧補間ドライブには、以下のドライブパラメータの設定が必要です。
・実行軸の加減速ドライブのパラメータ
・CIRCULAR XPOSITION　：現在位置の X 座標アドレス
・CIRCULAR YPOSITION　：現在位置の Y 座標アドレス
・CIRCULAR PULSE　：目的地の短軸座標までの短軸パルス数

任意軸補間ドライブのメイン軸円弧補間ドライブには、以下のドライブパラメータの設定が必要です。
・実行軸の加減速ドライブのパラメータ
・CP SPEC　：CPPOUT 出力
・CIRCULAR XPOSITION　：現在位置の X 座標アドレス
・CIRCULAR YPOSITION　：現在位置の Y 座標アドレス
・CIRCULAR PULSE　：目的地の短軸座標までの短軸パルス数

任意軸補間ドライブのサブ軸円弧補間ドライブには、以下のドライブパラメータの設定が必要です。
・CP SPEC　：CPPOUT 出力
・CIRCULAR XPOSITION　：現在位置の X 座標アドレス
・CIRCULAR YPOSITION　：現在位置の Y 座標アドレス
・CIRCULAR PULSE　：目的地の短軸座標までの短軸パルス数

■円弧補間ドライブの軌跡(CCW 回転の例)



円弧補間ドライブの軌跡は、現在位置と円弧の中心点の距離を半径とした円周に沿います。
・円弧補間 SCAN ドライブの場合は、停止指令を検出するまで指定の円弧半径と回転方向で
パルス出力を続けます。
・円弧補間 INDEX ドライブの場合は、短軸パルス数が指定の短軸パルス数になるとドライブを終了します。

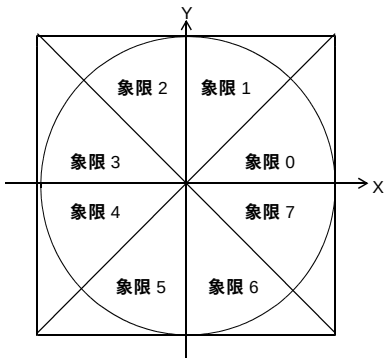
●円弧補間の短軸

円弧補間の中心点(0, 0)としたときに

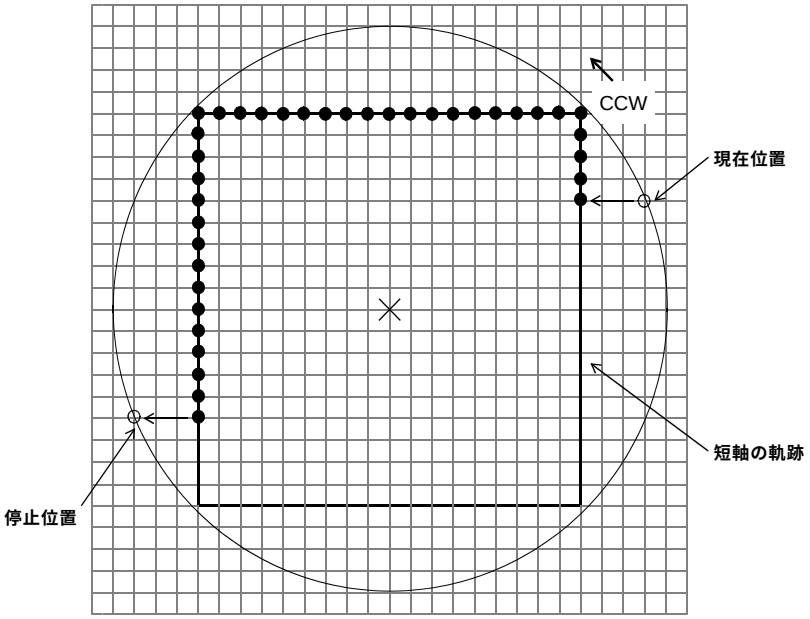
補間座標(X, Y)の絶対値が小さい方の軸が短軸になります。

右図の 1, 2, 5, 6 象限は X 軸が短軸です。

0, 3, 4, 7 象限は Y 軸が短軸です。



■円弧補間ドライブ短軸パルスの軌跡(CCW 回転の例)



円弧補間ドライブは、円弧の中心座標からみた短軸側が補間ドライブの基本パルス(短軸パルス)を
常に出力し、長軸側は基本パルス(短軸パルス)を補間演算して補間パルスを出力します。

●短軸パルス数の計算式
半径 R の円における１象限当たりの短軸パルス数 Ps は、以下の条件式で算出します。

K ＝ int(R／√2)　：int() は小数点以下を切り捨てた整数

(1)$R^2 \leqq K^2 + (K+1)^2$ のとき

条件式： $|K^2 + (K+1)^2 - R^2| > |K^2 + K^2 - R^2|$ のときは、Ps ＝ K
$|K^2 + (K+1)^2 - R^2| \leqq |K^2 + K^2 - R^2|$ のときは、Ps ＝ K ＋ 1/2

(2)$R^2 > K^2 + (K+1)^2$ のとき

条件式： $|K^2 + (K+1)^2 - R^2| > |(K+1)^2 + (K+1)^2 - R^2|$ のときは、Ps ＝ K ＋ 1
$|K^2 + (K+1)^2 - R^2| \leqq |(K+1)^2 + (K+1)^2 - R^2|$ のときは、Ps ＝ K ＋ 1/2

短軸パルス数 P ＝ int(各象限の短軸パルス数の合計)

1 象限当たりの短軸パルス数の 8 倍（Ps x 8）が、1 回転のパルス数になります。

●短軸パルス数の計算例 1（CCW 回転）
中心点（0, 0）に対して、現在位置を（10, 5）、目的地を（-10, -5）として CCW 回転させる場合の目的地の短軸座標までの短軸パルス数 P は、以下のようになります。

R ＝ $\sqrt{(10^2 + 5^2)}$ ＝ $\sqrt{125}$,　K ＝ int(R／√2) ＝ int(7.9) ＝ 7

R^2 ＝ 125,　$K^2 + (K+1)^2$ ＝ 113,　なので　$R^2 > K^2 + (K+1)^2$

$|K^2 + (K+1)^2 - R^2|$ ＝ 12,　$|(K+1)^2 + (K+1)^2 - R^2|$ ＝ 3, なので

$|K^2 + (K+1)^2 - R^2| > |(K+1)^2 + (K+1)^2 - R^2|$ のときは、Ps ＝ K ＋ 1 ＝ 8

短軸パルス数 P ＝ int(象限 0 のパルス数 ＋ 象限 1, 2, 3 のパルス数 ＋ 象限 4 のパルス数)
＝ (8 － 5) ＋ (8 ＋ 8 ＋ 8) ＋ 5 ＝ 32

CCW 回転は負数で指定するので、CIRCULAR PULSE ＝ -32 ＝ H'FFFF_FFE0

CCW 回転時の現在位置の象限の短軸パルス数は、以下のようになります。
・現在位置が象限 0, 2, 4, 6 の短軸パルス数： Ps －(現在位置の短軸座標の絶対値)
・現在位置が象限 1, 3, 5, 7 の短軸パルス数： 現在位置の短軸座標の絶対値

CCW 回転時の目的位置の象限の短軸パルス数は、以下のようになります。
・目的位置が象限 0, 2, 4, 6 の短軸パルス数： 目的位置の短軸座標の絶対値
・目的位置が象限 1, 3, 5, 7 の短軸パルス数： Ps －(目的位置の短軸座標の絶対値)

●短軸パルス数の計算例２（CW 回転）
中心点（0, 0）に対して、現在位置を（20, 5）、目的地を（-20, -5）として CW 回転させる場合の目的地の短軸座標までの短軸パルス数 P は、以下のようになります。

R ＝ $\sqrt{(20^2 + 5^2)}$ ＝ $\sqrt{425}$,　K ＝ int(R／√2) ＝ int(14.6) ＝ 14

R^2 ＝ 425,　$K^2 + (K+1)^2$ ＝ 421,　なので　$R^2 > K^2 + (K+1)^2$

$|K^2 + (K+1)^2 - R^2|$ ＝ 4,　$|(K+1)^2 + (K+1)^2 - R^2|$ ＝ 25,　なので

$|K^2 + (K+1)^2 - R^2| \leqq |(K+1)^2 + (K+1)^2 - R^2|$ のときは、Ps ＝ K ＋ 1/2 ＝ 14.5

短軸パルス数 P ＝ int(象限 0 のパルス数 ＋ 象限 7, 6, 5 のパルス数 ＋ 象限 4 のパルス数)
＝ 5 ＋ (14.5 ＋ 14.5 ＋ 14.5) ＋ (14.5 － 5) ＝ 58

CW 回転は正数で指定するので、CIRCULAR PULSE ＝ 58 ＝ H'0000_003A

CW 回転時の現在位置の象限の短軸パルス数は、以下のようになります。
・現在位置が象限 0, 2, 4, 6 の短軸パルス数： 現在位置の短軸座標の絶対値
・現在位置が象限 1, 3, 5, 7 の短軸パルス数： Ps －(現在位置の短軸座標の絶対値)

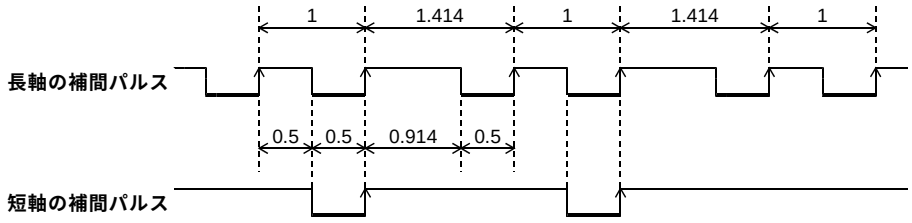
CW 回転時の目的位置の象限の短軸パルス数は、以下のようになります。
・目的位置が象限 0, 2, 4, 6 の短軸パルス数： Ps －(目的位置の短軸座標の絶対値)
・目的位置が象限 1, 3, 5, 7 の短軸パルス数： 目的位置の短軸座標の絶対値

(5) 線速一定制御

補間ドライブする 2 軸の合成速度を一定にする制御です。
コマンド実行軸が発生する補間ドライブの基本パルスを線速一定制御します。
コマンド実行軸に設定された速度が合成速度に反映されます。
2 軸同時にパルス出力したときに、次の基本パルスの出力周期を 1.414 倍にします。
・直線補間ドライブでは、コマンド実行軸の長軸と短軸の２軸間で、線速一定制御します。
・円弧補間ドライブでは、X 座標軸と Y 座標軸の２軸間で、線速一定制御します。
・線速一定制御は各補間ドライブの実行コマンドで設定します。

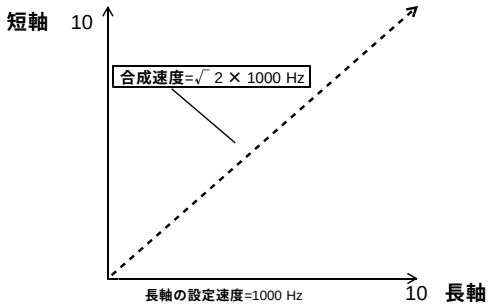
■線速一定の補間パルス出力 (２軸直線補間ドライブの例)

ON 周期の幅はそのままで、OFF 周期の幅が長くなります。



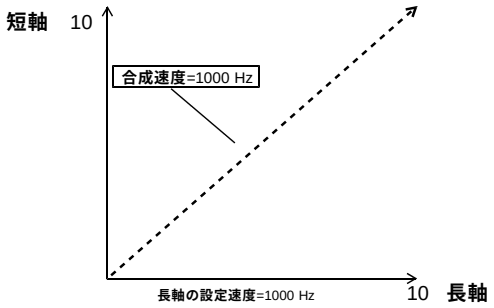
直線補間ドライブの軌跡(長軸 10:短軸 10 の例)

<線速一定制御なしのとき>



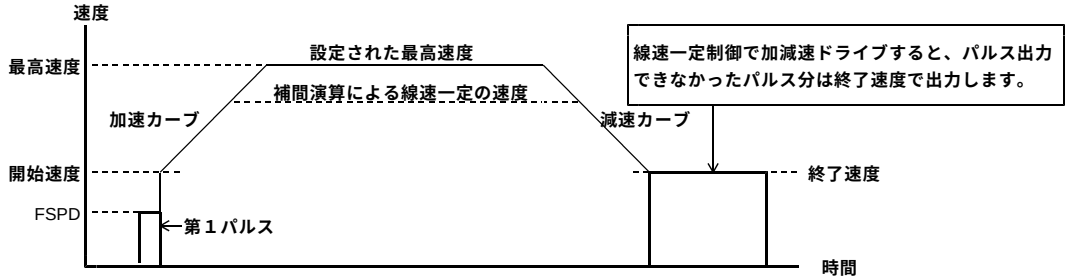
長軸の速度を一定速の 1000Hz とすると、
2 軸直線補間で描かれる軌跡の合成速度は
√ 2 × 1000Hz でドライブします。

<線速一定制御ありのとき>



コマンド実行軸の速度を一定速の 1000Hz として
線速一定制御を設定すると、2 軸直線補間で
描かれる軌跡の合成速度が 1000Hz となるように
ドライブします。

※線速一定で加減速ドライブを行うと、減速後の終了速度でのドライブが長くなります。



【注意事項】
線速一定制御には注意事項があります。

線速一定制御有効の円弧補間ドライブが、2 軸同時にパルス出力した位置で終了した場合に、
以降に実行する線速一定制御有効の直線補間ドライブのパルス出力が、
常に設定値の 1.414 倍の周期（常に線速一定制御される）になります。
・1 軸のみパルス出力する位置(例: 0 °,90 °,180 °,270 °)で終了した場合は正常です。
・2 軸同時にパルス出力する位置(例:45 °,135 °,225 °,315 °)で終了した場合に不具合が発生します。

線速一定制御有効の円弧補間ドライブ終了後は、
以下の円弧補間ドライブ（0 パルス、終了位置 0 °）を実行して、正常終了にしてください。
・CIRCULAR XPOSITION SET コマンド (H'28) ：H'00_0000 に設定
・CIRCULAR YPOSITION SET コマンド (H'29) ：H'00_0000 に設定
・CIRCULAR PULSE SET コマンド (H'2A) ：H'0000_0000 に設定
・MAIN CIRCULAR CP コマンド (H'38) ：DATA1=H'0001 で実行

4-1-6. ドライブ CHANGE 機能

ドライブ実行中に、各種ドライブ CHANGE 指令を実行することができます。

(1) UP/DOWN/CONST ドライブ CHANGE 機能

UP/DOWN/CONST のドライブ CHANGE 指令は、STBY = 1 から有効になります。
変更動作点の検出で、UP/DOWN/CONST のドライブ CHANGE 指令を実行します。

停止指令またはエラーを検出すると、ドライブ CHANGE 指令は無効になります。
INDEX ドライブの減速地点を検出すると、ドライブ CHANGE 指令は無効になります。

UP DRIVE 指令を検出すると、最高速度まで加速または減速します。

DOWN DRIVE 指令を検出すると、終了速度まで加速または減速します。

CONST DRIVE 指令を検出すると、加速または減速を終了して、一定速にします。
・直線加減速ドライブの加減速中の場合は、CHANGE 指令の検出で加速または減速を終了します。
・S 字加減速ドライブの加減速中の場合は、S 字カーブで滑らかに加速または減速を終了します。

UP/DOWN/CONST のドライブ CHANGE 指令には、以下のドライブパラメータの設定が必要です。
・UDC SPEC ：UP/DOWN/CONST のドライブ CHANGE 指令を実行する変更動作点

■ UP/DOWN/CONST ドライブ指令が有効となるコマンド

ORIGIN ドライブでは、加減速の SCAN ドライブ実行時に有効です。
補間ドライブでは、実行軸の加減速パラメータに対して CHANGE 指令を実行します。

COMMAND CODE	汎用コマンド名称		機能
H'12	+SCAN	*P	＋方向 SCAN ドライブの実行
H'13	-SCAN	*P	－方向 SCAN ドライブの実行
H'14	INC INDEX	*P	相対アドレス INDEX ドライブの実行
H'15	ABS INDEX	*P	絶対アドレス INDEX ドライブの実行
H'18	ORIGIN SCAN	*P	ORIGIN SCAN ドライブの実行
H'32	MAIN XY STRAIGHT CP	*P	相関 2 軸直線補間ドライブの実行
H'3A	MAIN XY CIRCULAR CP	*P	相関 2 軸円弧補間ドライブの実行
H'30	MAIN STRAIGHT CP	*P	任意軸補間のメイン軸直線補間ドライブの実行
H'38	MAIN CIRCULAR CP	*P	任意軸補間のメイン軸円弧補間ドライブの実行

・上記のドライブを実行すると、STBY = 1 で SPEED CBUSY = 0 になります。
　上記のドライブ以外の実行では、SPEED CBUSY = 1 のままです。

・ドライブ CHANGE 指令の書き込みで、SPEED CBUSY = 1、SPEED CSET = 1 になります。
・ドライブ CHANGE 指令の実行終了で、SPEED CBUSY = 0、SPEED CSET = 0 になります。

・以下の状態を検出すると、SPEED CBUSY = 1、SPEED CSET = 0（クリア）になります。
・INDEX ドライブ、ORIGIN SCAN ドライブの減速地点の検出
・LSEND = 1、SSEND = 1、FSEND = 1、ERROR = 1 の検出
　SPEED CSET をクリアした場合は、実行待ちのドライブ CHANGE 指令は無効になります。

■ UP/DOWN/CONST ドライブ CHANGE 信号

SS0 信号の操作で、UP/DOWN/CONST のドライブ CHANGE ができます。
SS0 信号のドライブ CHANGE では、変更動作点の設定は無効になります。

SS0 信号のドライブ CHANGE 機能は、SPEC INITIALIZE2 コマンドで設定します。
・SS0 信号は下記の軸に対応しています。Z 軸および A 軸には対応していません。

/IN0 信号	X 軸の SS0 信号
/IN1 信号	Y 軸の SS0 信号

SS0 信号のレベル変化の検出で、ドライブ CHANGE 指令を実行します。
・SPEC INITIALIZE2 で設定した CHANGE 機能で SS0 信号による UP DRIVE, DOWN DRIVE, CONST DRIVE のドライブ CHANGE 指令が実行できます。

● SS0 信号によるドライブ CHANGE 動作
ORIGIN ドライブでは、加減速の SCAN ドライブ実行時に有効です。
補間ドライブでは、メイン軸の加減速パラメータに対して CHANGE 指令を実行します。

・信号のレベル変化の検出は、SPEED CBUSY = 0 となるドライブの STBY = 1 から有効になります。
STBY = 1 で検出した信号のレベルがドライブ CHANGE 指令の場合は、STBY = 1 からドライブ CHANGE 指令を開始します。

・ドライブ CHANGE 指令を検出すると、SPEED CBUSY = 1、SPEED CSET = 1 になります。
SPEED CBUSY = 1 の間は、信号のレベル変化の検出は保留になります。

・ドライブ CHANGE 指令の実行終了で、SPEED CBUSY = 0、SPEED CSET = 0 になります。
同時に、ドライブ CHANGE 信号のレベル変化の検出が有効になります。

・ドライブ CHANGE 指令実行後に、実行前と同じ信号レベル（または機能なし状態）を検出した場合は、SPEED CBUSY = 0、SPEED CSET = 0 のままです。
ドライブ CHANGE 指令実行後に、実行前と異なる信号レベル（異なるドライブ CHANGE 指令）を検出した場合は、SPEED CBUSY = 1、SPEED CSET = 1 になり、次のドライブ CHANGE 指令を実行します。

・以下の状態を検出すると、SPEED CBUSY = 1、SPEED CSET = 0（クリア）になります。
  ・INDEX ドライブ、ORIGIN SCAN ドライブの減速地点の検出
  ・LSEND = 1、SSEND = 1、FSEND = 1、ERROR = 1 の検出
SPEED CSET をクリアした場合は、ドライブ CHANGE 指令は無効になります。

(2) SPEED CHANGE 機能

SPEED CHANGE 指令は、STBY = 1 から有効になります。
変更動作点の検出で、SPEED CHANGE 指令を実行します。

停止指令またはエラーを検出すると、SPEED CHANGE 指令は無効になります。
INDEX ドライブの減速地点を検出すると、SPEED CHANGE 指令は無効になります。

SPEED CHANGE 指令を検出すると、指定したドライブパルス速度まで加速または減速します。
指定する速度は、最高速度以上および開始速度／終了速度以下にできます。

- STBY = 1 で SPEED CHANGE 指令を検出した場合は、最初の加速の目標速度を SPEED CHANGE 指令の指定速度にします（開始速度＜最高速度、開始速度＜ SPEED CHANGE 指定速度の場合）。
- 直線加減速ドライブの加減速中に SPEED CHANGE 指令を検出した場合は、指令の検出で指定速度まで加速または減速します。
- S 字加減速ドライブの加減速中に SPEED CHANGE 指令を検出した場合は、現在の加減速状態を S 字カーブで滑らかに終了させてから、指定速度まで加速または減速します。

SPEED CHANGE 指令によるドライブパルス速度の変更は、現在のドライブ中のみの変更です。
SPEED CHANGE 指令を実行しても、速度パラメータの設定は変わりません。

SPEED CHANGE 指令には、以下のドライブパラメータの設定が必要です。

- SPEED CHANGE SPEC ：SPEED CHANGE 指令を実行する変更動作点

■ SPEED CHANGE 指令が有効となるコマンド

ORIGIN ドライブでは、加減速の SCAN ドライブ実行時に有効です。
補間ドライブでは、実行軸の加減速パラメータに対して CHANGE 指令を実行します。

H'12	+SCAN	*P	＋方向 SCAN ドライブの実行
H'13	-SCAN	*P	－方向 SCAN ドライブの実行
H'14	INC INDEX	*P	相対アドレス INDEX ドライブの実行
H'15	ABS INDEX	*P	絶対アドレス INDEX ドライブの実行
H'18	ORIGIN SCAN	*P	ORIGIN SCAN ドライブの実行
H'32	MAIN XY STRAIGHT CP	*P	相関 2 軸直線補間ドライブの実行
H'3A	MAIN XY CIRCULAR CP	*P	相関 2 軸円弧補間ドライブの実行
H'30	MAIN STRAIGHT CP	*P	任意軸補間のメイン軸直線補間ドライブの実行
H'38	MAIN CIRCULAR CP	*P	任意軸補間のメイン軸円弧補間ドライブの実行

- 上記のドライブを実行すると、STBY = 1 で SPEED CBUSY = 0 になります。
  上記のドライブ以外の実行では、SPEED CBUSY = 1 のままです。
- SPEED CHANGE 指令の書き込みで、SPEED CBUSY = 1、SPEED CSET = 1 になります。
- SPEED CHANGE 指令の実行終了で、SPEED CBUSY = 0、SPEED CSET = 0 になります。
- 以下の状態を検出すると、SPEED CBUSY = 1、SPEED CSET = 0（クリア）になります。
  INDEX ドライブ、ORIGIN SCAN ドライブの減速地点の検出
  LSEND = 1、SSEND = 1、FSEND = 1、ERROR = 1 の検出
  SPEED CSET をクリアした場合は、実行待ちのドライブ CHANGE 指令は無効になります。

(3) RATE CHANGE 機能

RATE CHANGE 指令は、STBY = 1 から有効になります。
RATE CHANGE 指令は、以下のドライブ CHANGE 指令の検出と同時に実行します。
・UP DRIVE、DOWN DRIVE、CONST DRIVE、SPEED CHANGE

停止指令またはエラーを検出すると、RATE CHANGE 指令は無効になります。
INDEX ドライブの減速地点を検出すると、RATE CHANGE 指令は無効になります。

RATE CHANGE 指令は、ドライブ CHANGE 動作時の変速周期データの変更です。
RATE CHANGE 指令を検出すると、ドライブ CHANGE 動作時の加速カーブと減速カーブの変速周期データを、指定したデータに変更します。
・RATE CHANGE 指令を設定すると、他のドライブ CHANGE 指令の検出と同時に RATE CHANGE 指令を検出し、変速周期データを変更します。
・変更した変速周期データは、次のドライブ CHANGE 動作時も有効です。
現在のドライブが終了すると、変更した変速周期データは無効になります。
・減速停止動作時は、RATE SET コマンドで設定した DCYCLE の変速周期で減速停止します。

RATE CHANGE 指令による変速周期データの変更は、ドライブ CHANGE 動作時のみの変更です。
RATE CHANGE 指令を実行しても、速度パラメータの設定は変わりません。

■ RATE CHANGE 指令が有効となるコマンド

ORIGIN ドライブでは、加減速の SCAN ドライブ実行時に有効です。
補間ドライブでは、実行軸の加減速パラメータに対して CHANGE 指令を実行します。

H'12	+SCAN	*P	＋方向 SCAN ドライブの実行
H'13	-SCAN	*P	－方向 SCAN ドライブの実行
H'14	INC INDEX	*P	相対アドレス INDEX ドライブの実行
H'15	ABS INDEX	*P	絶対アドレス INDEX ドライブの実行
H'18	ORIGIN SCAN	*P	ORIGIN SCAN ドライブの実行
H'32	MAIN XY STRAIGHT CP	*P	相関 2 軸直線補間ドライブの実行
H'3A	MAIN XY CIRCULAR CP	*P	相関 2 軸円弧補間ドライブの実行
H'30	MAIN STRAIGHT CP	*P	任意軸補間のメイン軸直線補間ドライブの実行
H'38	MAIN CIRCULAR CP	*P	任意軸補間のメイン軸円弧補間ドライブの実行

・上記のドライブを実行すると、STBY = 1 で SPEED CBUSY = 0 になります。
上記のドライブ以外の実行では、SPEED CBUSY = 1 のままです。

・RATE CHANGE 指令の書き込みで、RATE CSET = 1 になります。
RATE CSET = 1 でも、RATE CHANGE 指令の書き込みは可能です（上書きします）。
（他のドライブ CHANGE 指令の書き込みで、SPEED CBUSY = 1、SPEED CSET = 1 になります）

・他のドライブ CHANGE 指令の実行と同時に RATE CHANGE 指令を実行し、RATE CSET = 0 になります。
（他のドライブ CHANGE 指令の実行終了で、SPEED CBUSY = 0、SPEED CSET = 0 になります）

・以下の状態を検出すると、SPEED CBUSY = 1、RATE CSET = 0（クリア）になります。
・INDEX ドライブ、ORIGIN SCAN ドライブの減速地点の検出
・LSEND = 1、SSEND = 1、FSEND = 1、ERROR = 1 の検出
RATE CSET をクリアした場合は、実行待ちの RATE CHANGE 指令は無効になります。

(4) INDEX CHANGE 機能

INDEX CHANGE 指令は、STBY = 1 から有効になります。
変更動作点の検出で、INDEX CHANGE 指令を実行します。

停止指令またはエラーを検出すると、INDEX CHANGE 指令は無効になります。
・実行待ちの INDEX CHANGE 指令が無効になった場合は、エラーになります。
ERROR STATUS の CHANGE CLR ERROR = 1 にします。

・反転動作が必要な INDEX CHANGE 指令を検出した場合は、エラーになります。
実行中のドライブを減速停止して、ERROR STATUS の INDEX CHANGE ERROR = 1 にします。

INDEX CHANGE 指令は、INC/ABS/PLS INDEX CHANGE の３種類あります。
・INC INDEX CHANGE 指令を検出すると、指定したデータを、起動位置を原点とする相対アドレスの停止位置に設定して、INC INDEX ドライブを行います。
＊ INC INDEX CHANGE コマンドの指定データは、実行中のドライブ方向と同じ符号にしてください。
詳しくは、「INC INDEX CHANGE コマンド」の頁をご覧ください。

・ABS INDEX CHANGE 指令を検出すると、指定したデータを、アドレスカウンタで管理している絶対アドレスの停止位置に設定して、ABS INDEX ドライブを行います。

・PLS INDEX CHANGE 指令を検出すると、指定したデータを、変更動作点の検出位置を原点とする相対アドレスの停止位置に設定して、INC INDEX ドライブを行います。

通常の INDEX ドライブでは、自動減速停止動作開始後に、停止位置を検出した時点で停止します。
INDEX CHANGE 指令を検出した場合は、自動減速停止動作開始後に、終了速度に達してから停止位置を検出して停止します。

INDEX CHANGE 指令には、以下のドライブパラメータの設定が必要です。
・INDEX CHANGE SPEC ：INDEX CHANGE 指令を実行する変更動作点

■ INDEX CHANGE 指令が有効となるドライブ（コマンド）

H'12	+SCAN	*P	＋方向 SCAN ドライブの実行
H'13	-SCAN	*P	－方向 SCAN ドライブの実行
H'14	INC INDEX	*P	相対アドレス INDEX ドライブの実行
H'15	ABS INDEX	*P	絶対アドレス INDEX ドライブの実行

・上記のドライブを実行すると、STBY = 1 で INDEX CBUSY = 0 になります。
上記のドライブ以外の実行では、INDEX CBUSY = 1 のままです。

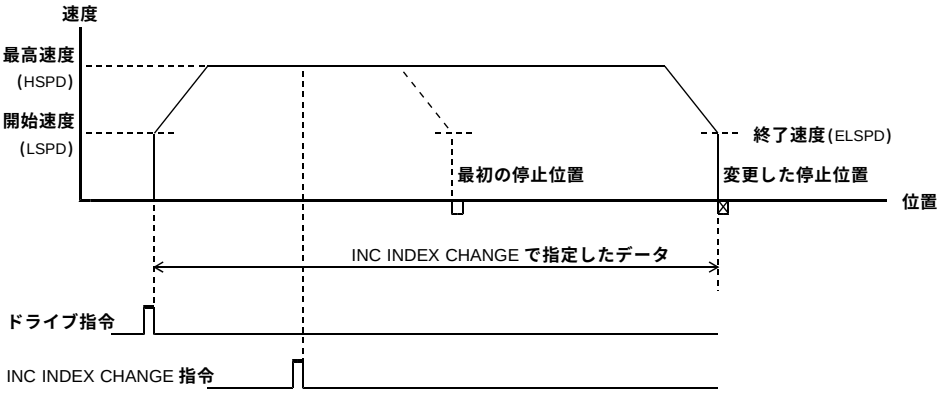
・INDEX CHANGE 指令の書き込みで、INDEX CBUSY = 1、INDEX CSET = 1 になります。
・INDEX CHANGE 指令の実行終了で、INDEX CBUSY = 0、INDEX CSET = 0 になります。
・ドライブが終了すると、DRIVE = 0 で INDEX CBUSY = 1、INDEX CSET = 0 になります。

・以下の状態を検出すると、INDEX CBUSY = 1、INDEX CSET = 0（クリア）になります。
LSEND = 1、SSEND = 1、FSEND = 1、ERROR = 1 の検出
INDEX CSET をクリアした場合は、実行待ちの INDEX CHANGE 指令は無効になります。

・実行待ちの INDEX CHANGE 指令が無効になった場合は、ERROR = 1 になります。

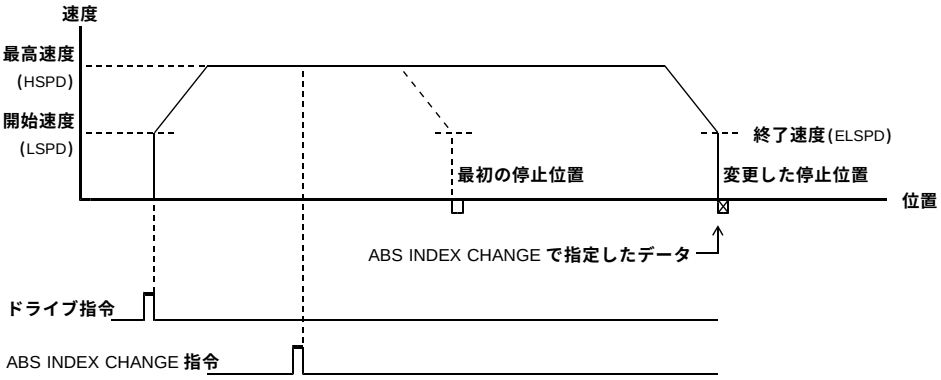
■ INC INDEX CHANGE の動作

指定したデータを、起動位置を原点とする相対アドレスの停止位置にします。



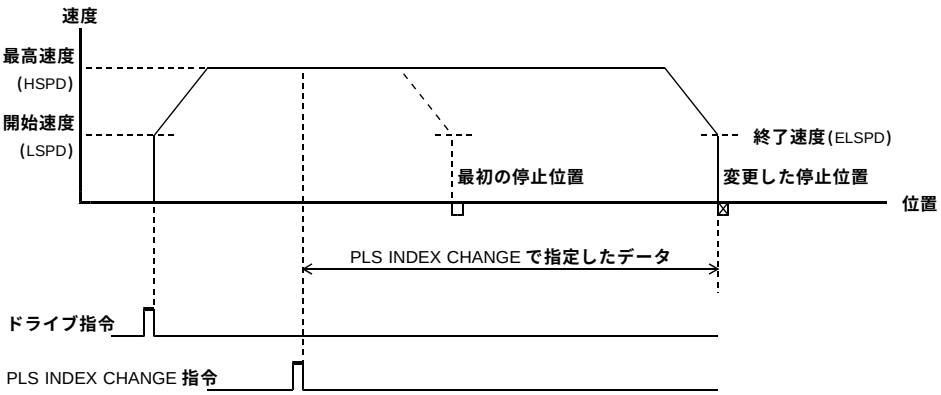
■ ABS INDEX CHANGE の動作

指定したデータを、アドレスカウンタで管理している絶対アドレスの停止位置にします。



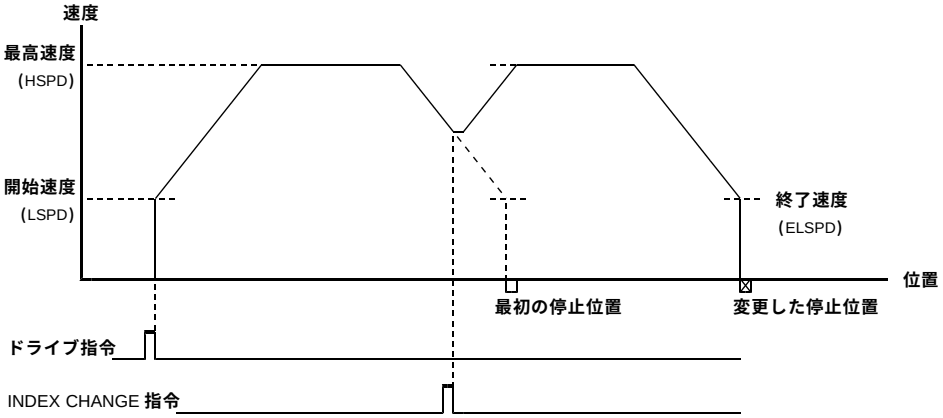
■ PLS INDEX CHANGE の動作

指定したデータを、変更動作点の検出位置を原点とする相対アドレスの停止位置にします。

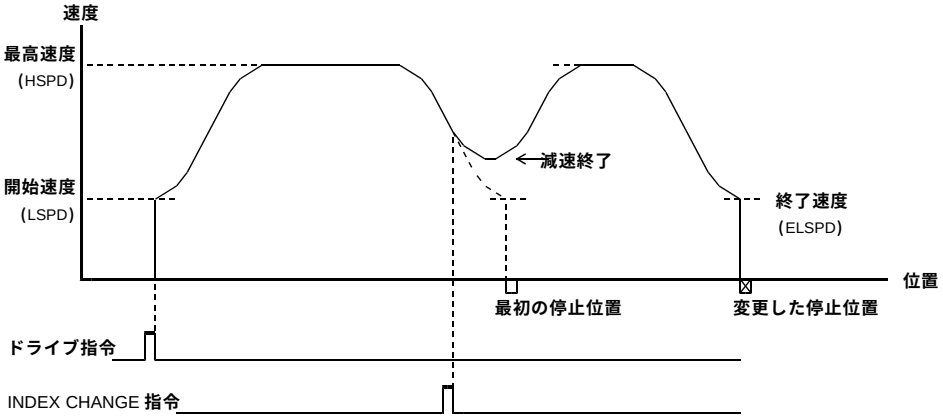


■減速中の INDEX CHANGE 動作

●直線減速中の INDEX CHANGE
直線加減速ドライブでは、停止位置への減速中に、加速が必要な INDEX CHANGE 指令を検出した場合は、減速の途中から再加速して、変更した停止位置までドライブします。



● S 字減速中の INDEX CHANGE
S 字加減速ドライブでは、停止位置への減速中に、加速が必要な INDEX CHANGE 指令を検出した場合は、S 字減速カーブで滑らかに減速を終了させてから、S 字加速カーブで再加速します。



4-1-7. 読み出し機能

下記データの読み出しは、取扱説明書(標準)の 3-7.章の制限事項を除き、常時可能です。

下記の関数にて、コマンドの書き込みと、データの読み出しを一括で処理することができます。
これらの関数は、コマンドの書き込みからデータの読み出しまで、関数内で排他処理されています。
マルチスレッドプログラミングのように、複数のスレッドで処理される場合は、この関数によって
読み出しすることを推奨します。
・デバイス単位 … DRIVE COMMAND 32 ビット書き込み／読み出し関数

(1) カウントデータのラッチデータ読み出し

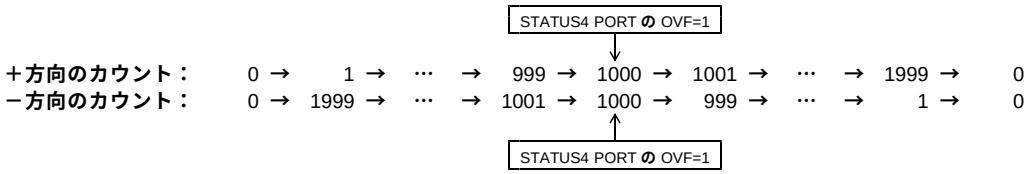
MCC07 に各カウンタ LATCH DATA READ コマンドを実行するとラッチしたカウントデータを
読み出すことができます。

4-2. カウンタ仕様

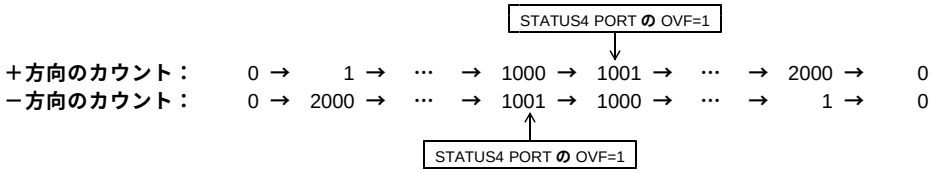
4-2-1. リングカウンタ機能

アドレスカウンタ、およびパルスカウンタは、カウント最大値を設定すると、設定値をカウンタの最大値としてリングカウントします。
回転系のアドレス管理に便利です。

●最大カウント数＝ 1,999 の場合(2,000 カウントで 1 回転)



●最大カウント数＝ 2,000 の場合(2,001 カウントで 1 回転)



・パルス偏差カウンタにはリングカウンタ機能はありません。

・カウント数が設定値の 1/2 に達すると、STATUS4 PORT の各カウンタの OVF=1 になります。

・カウント最大値は各カウンタの COUNTER MAX COUNT SET コマンドで設定します。

・アドレスカウンタにカウント最大値を設定して、ABS INDEX ドライブでカウント最大値以上のアドレスを指定した場合、アドレスカウンタが OVF した時点でドライブは停止し、エラーとなります。

4-2-2. カウントデータのラッチ・クリア機能

■カウンタのラッチ機能

設定したラッチタイミングのアクティブエッジで、カウンタのカウントデータをラッチします。
ラッチしたデータは次のラッチタイミングのアクティブエッジを検出するまで保存します。
ラッチしたデータは各カウンタ LATCH DATA READ コマンドで読み出すことができます。

●設定できるラッチタイミング

ラッチタイミング <エッジ検出>
・各カウンタ LATCH DATA READ コマンドの実行でラッチする
・多用途センサ信号：SS0 = 0 → 1 でラッチする
・ORIGIN SPEC SET コマンドの ORG 検出信号の検出エッジでラッチする

・各カウンタのラッチタイミングは COUNT LATCH SPEC SET コマンドで設定します。

(注) SS0 信号は SPEC INITIALIZE2 コマンドで「各種トリガ」に設定している場合に有効です。

・SS0 信号は下記の軸に対応しています。Z 軸および A 軸には対応していません。

/IN0 信号	X 軸の SS0 信号
/IN1 信号	Y 軸の SS0 信号

■カウンタのクリア機能

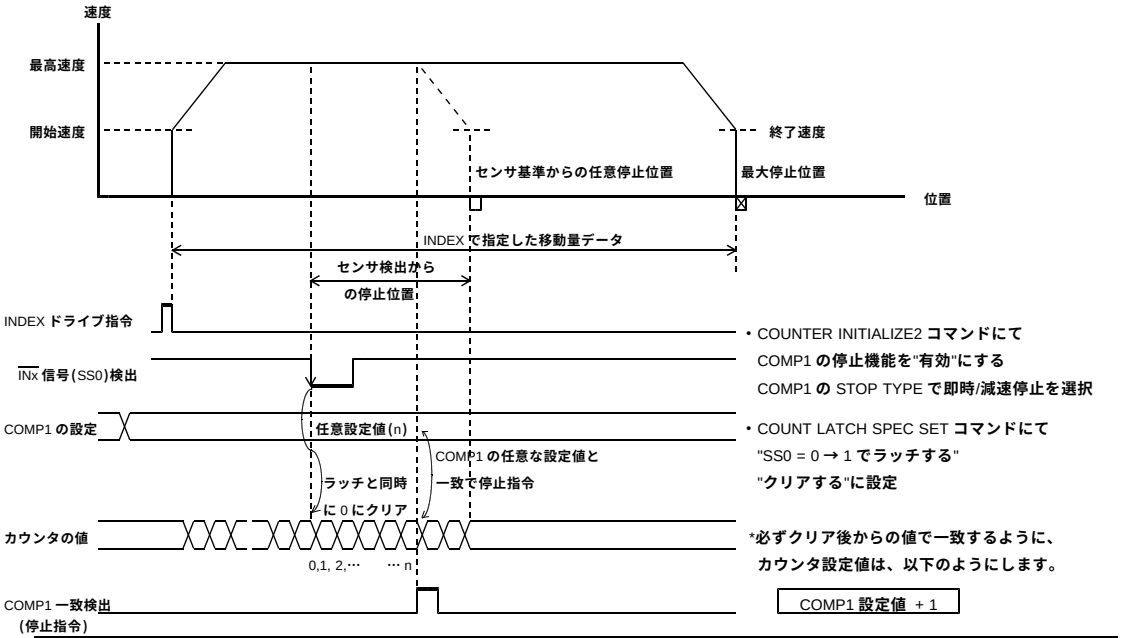
カウントデータのラッチと同時にカウントデータを "0" にクリアします。
カウンタのカウントタイミングとクリアタイミングが同時に発生した場合はクリアを優先します。

・各カウンタのラッチクリア機能は COUNT LATCH SPEC SET コマンドで設定します。

●カウンタラッチとクリア機能を応用したドライブ例

センサ(IN0 信号または IN1 信号)の検出を基準とした位置から、カウンタのコンパレーター一致による停止機能でドライブを停止させることが可能です。

・ドライブ起動前に必要な設定を済ませることができ、ドライブ中の Windows OS や AL-Ⅱ通信などの影響を受けない繰り返し精度の高いセンサ検出によるドライブが可能です。
・予め INDEX ドライブを起動することで、センサ信号検出しない場合でもメカの最大移動量は保証できます。
・センサ検出から減速開始または即時停止するまでの移動量をコンパレータの設定値により調整できます。減速停止時の減速開始から出力されるパルス数は保証されます。
・ドライブ中の他の停止信号や停止コマンドも有効です。



4-3. I/0 仕様

4-3-1. その他の I/O PORT

(1) SIGNAL I/O 信号

2C-776Av1 は、X 軸および Y 軸のステータス信号(SOUT 信号)を、SIGNAL I/O コネクタの SOUT 信号から、外部にステータス信号を出力することができます。
出力信号は接続する信号によって、オープンコレクタまたはラインドライバ出力が選択できます。
・X 軸の MCC SOUT 信号(初期値 CNTINT)は、SOUT0 信号から出力されます。
・Y 軸の MCC SOUT 信号(初期値 CNTINT)は、SOUT1 信号から出力されます。

外部出力信号	出力できる信号
$\overline{\text{SOUT0}}$	X 軸の SOUT 信号
$\overline{\text{SOUT1}}$	Y 軸の SOUT 信号

SOUT 信号の状態は、X 軸または Y 軸の DRIVE STATUS3 PORT から読み出すことができます。

■ SOUT 信号

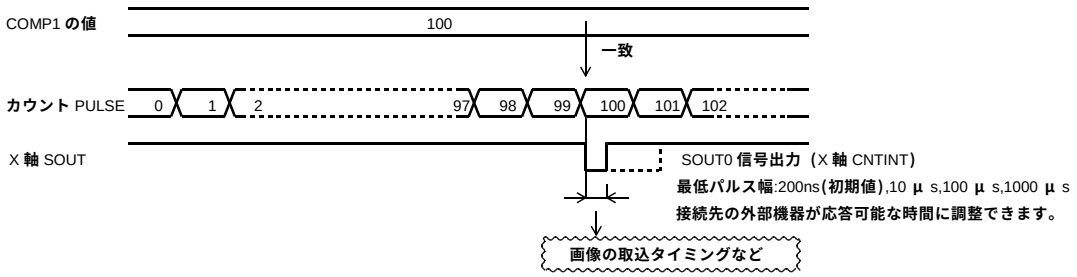
HARD INITIALIZE1 コマンドにより、下記の中から出力する信号を選択します。

- カウンタ割り込み要求の ARDINT
- カウンタ割り込み要求の CNTINT
- カウンタ割り込み要求の DFLINT
- コマンド終了割り込みの RDYINT
- STATUS1 の STBY フラグ
- STATUS1 の DRIVE フラグの反転
- STATUS5 の SPEED CBUSY の反転
- STATUS5 の INDEX CBUSY の反転
- STATUS1 の UP フラグ
- STATUS1 の DOWN フラグ
- STATUS1 の CONST フラグ
- STATUS1 の EXT PULSE フラグ
- STATUS2 の PULSE MASK フラグの反転
- STATUS2 の ORG SIGNAL フラグ
- 汎用出力としての出力状態
- STATUS4 の PULSE OVF フラグ

■ カウンター致信号で外部機器との同期を行う例

・X 軸の SOUT 機能を HARD INITIALIZE1 コマンドで CNTINT に設定します。(初期値)
・PULSE COUNTER INITIALIZE2 コマンドで COMP(1,2,3)の一致出力を「CNTINT に出力する」に設定します。
この COMP 条件が一致すると、SOUT 信号が ON します。

【パルスカウンタ値 100 で一致出力を行った例】



●その他

・一致出力と同時にカウンタの値を自動的にクリアする AUTO CLEAR 機能があります。
アプリーションから毎回コンパレータ設定する必要がなく、相対値に達する度に一致出力ができます。
このときの一致出力も最小パルス幅で SOUT 信号が ON します。

・一致出力と同時にカウンタの値を COMP1 ADD で設定された値に自動加算する AUTO ADD 機能があります。
アプリーションから毎回コンパレータ設定する必要がなく、絶対値で管理しながら、
COMPARE REGISTER1 ＝ COMPARE REGISTER1 ＋ COMP1 ADD の値で一致出力ができます。

5．付録

5-1.初期仕様一覧

(1) 応用設定

項　目	初期仕様	対応関数/コマンド
■入力信号のデジタルフィルタ機能		
CWLM, CCWLM 信号	0 ～ 50ns	HARD INITIALIZE4 コマンド
DEND/PO, DALM 信号	0 ～ 50ns	
ORG, NORG 信号	0 ～ 50ns	HARD INITIALIZE5 コマンド
± ZORG 信号	0 ～ 50ns	
± EA, ± EB 信号	0 ～ 50ns	HARD INITIALIZE6 コマンド
■入力信号のアクティブ論理		
FSSTOP 信号	正論理	HARD INITIALIZE7 コマンド
CWLM 信号	正論理	
CCWOM 信号	正論理	
DALM 信号	正論理	
PAUSE 信号	1 でアクティブ	
■アドレスカウンタ		
最大カウント数	H'FFFF_FFFF	ADDRESS COUNTER MAX COUNT SET コマンド
■パルスカウンタ		
最大カウント数	H'FFFF_FFFF	PULSE COUNTER MAX COUNT SET コマンド
■カウンタのラッチ・クリア機能		
アドレスカウンタラッチタイミング	ADDRESS LATCH DATA READ コマンドの実行	COUNT LATCH SPEC SET コマンド
ADDRESS CLR ENABLE	クリアしない	
パルスカウンタラッチタイミング	PULSE LATCH DATA READ コマンドの実行	
PULSE CLR ENALBE	クリアしない	
パルス偏差カウンタラッチタイミング	DFL LATCH DATA READ コマンドの実行	
DFL CLR ENABLE	クリアしない	
■センサ機能		
X 軸 SS0 入力機能	汎用入力、各種機能のトリガ入力	SPEC INITIALIZE2 コマンド
Y 軸 SS0 入力機能	汎用入力、各種機能のトリガ入力	
■ステータス外部出力機能		
X 軸 SOUT 出力ステータス	CNTINT	HARD INITIALIZE1 コマンド
Y 軸 SOUT 出力ステータス	CNTINT	

(2) 応用ドライブパラメータ

項　目	初期仕様	対応関数/コマンド
■第 1 パルス出力周期　(FSPD)	5,000Hz　(=200 μ s/1 周期)	FSPD SET コマンド
■加減速パラメータ		
速度倍率　(RESOL)	1 (No.3)	HIGH SPEED SET コマンド
最高速時の速度データ　(HSPD)	3,000	
加速開始時の速度データ　(LSPD)	300	LOW SPEED SET コマンド
減速終了時の速度データ　(ELSPD)	300	
加速カーブ変速周期データ　(UCYCLE)	200	RATE SET コマンド
減速カーブ変速周期データ　(DCYCLE)	200	
加速カーブ S 字変速領域データ　(SUAREA)	0(変速領域なし)	SCAREA SET コマンド
減速カーブ S 字変速領域データ　(SDAREA)	0(変速領域なし)	
減速パルス数のオフセットパルス数	1 パルス	DOWN PULSE ADJUST コマンド
■ ORIGIN ドライブパラメータ		
ORG 検出信号	$\overline{\text{ORG}}$ 信号と± ZORG 信号の OR(論理和)	ORIGIN SPEC SET コマンド
ORG 検出信号の検出エッジ	ORG 検出信号の 0→1(アクティブ)エッジ	
ORG ドライブ起動方向	－(CCW)方向	
検出エッジのカウント数	1 カウント目のエッジ検出で停止	
停止時 DRST 出力	出力しない	
■補間ドライブパラメータ		
CPPOUT 端子から出力するパルス	CPPIN 端子から入力するパルス	CP SPEC SET コマンド
直線補間 長軸の目的地の座標アドレス	H'0000_0000	LONG POSITION SET コマンド
直線補間 短軸の目的地の座標アドレス	H'0000_0000	SHORT POSITION SET コマンド
円弧補間 現在位置の X 座標アドレス	H'00_0000	CIRCULAR XPOSITION SET コマンド
円弧補間 現在位置の Y 座標アドレス	H'00_0000	CIRCULAR YPOSITION SET コマンド
円弧補間 目的地までの短軸パルス数	H'0000_0000	CIRCULAR PULSE SET コマンド
■ドライブ CHANGE パラメータ		
UP DRIVE 変更動作点	UP DRIVE CHANGE コマンドの実行	UDC SPEC SET コマンド
DOWN DRIVE 変更動作点	DOWN DRIVE CHANGE コマンドの実行	
CONST DRIVE 変更動作点	CONST DRIVE CHANGE コマンド実行	
SPEED CHANGE 変更動作点	SPEED CHANGE コマンドの実行	SPEED CHANGE SPEC SET コマンド
INDEX CHANGE 変更動作点	INDEX CHANGE コマンドの実行	INDEX CHANGE SPEC SET コマンド

5-2. 関数一覧

種別	名称 / 説明	記号	ページ 標準編	ページ 応用編
共通構造体	RESULT構造体 関数を実行した結果を格納する	MC07_S_RESULT	32	
	コマンドデータ構造体 DRIVE COMMAND PORT、DRIVE DATA1 PORT。DRIVE DATA2 PORTに書き込むデータを格納する	MC07_S_COMMAND_DATA	35	
	ステータスデータ構造体 DRIVE STATUS1 PORT、DRIVE DATA1 PORT、DRIVE DATA2 PORTから読み出した内容を格納する	MC07_S_STATUS_DATA	36	
インフォメーション	スレーブ情報構造体 AL-Ⅱ通信に接続される全スレーブユニットのタイプを格納する	MC07_S_SLAVE_INFO	37	
	環境設定関数 AL-Ⅱ通信に接続されている全スレーブユニットを対象に環境設定を行う	MC07_Environment	39	
	スレーブ情報読み出し関数 AL-Ⅱ通信に接続されている全スレーブユニットのタイプを読み出す	MC07_ReadUnitInfo	40	
	AL-Ⅱ通信エラー累計回数読み出し関数 AL-Ⅱ通信のエラー累計回数を読み出す	MC07_ErrCount	41	
	AL-Ⅱ通信エラー累計回数クリア関数 AL-Ⅱ通信のエラー累計回数を0にクリアする	MC07_ClrErrCount	42	
ユニット	ユニットオープン関数 指定ユニット番号でユニットオープンし、引数*phUnit*の変数にユニットハンドルを格納する	MC07_UOpen	43	
	ユニットクローズ関数 指定されたユニットをクローズする	MC07_UClose	44	
	ユニット動作エラークリア関数 指定ユニットに対し、指定された軸の動作エラークリア処理を一括で行う	MC07_UClrError	45	
	拡張ユニット通信設定関数 指定されたユニットと拡張ユニット間の通信設定を行う	MC07_UWExUnitCommMode	46	
	拡張ユニット通信制御関数 指定されたユニットと拡張ユニット間の通信を制御する	MC07_UWExUnitCommControl	47	
	拡張ユニット通信ステータス読み出し関数 指定されたユニットと拡張ユニット間の通信の状態を読み出す	MC07_URExUnitCommStatus	48	
	拡張ユニット通信設定読み出し関数 指定されたユニットと拡張ユニット間の通信設定を読み出す	MC07_URExUnitCommMode	50	
	拡張ユニットGI/Oユニット通信制御関数 指定されたスレーブGユニットと拡張GI/Oユニット間の通信を制御する	MC07_UWGExUnitCommControl	51	
	拡張ユニットGI/Oユニット通信ステータス読み出し関数 指定されたユニットと拡張GI/Oユニット間の通信の状態を読み出す	MC07_URGExUnitCommStatus	52	
ステータスユニット	ユニットステータス構造体 ユニットのステータスの内容を格納する	MC07_S_UNIT_STATUS	54	
	ユニットコマンド構造体 ユニットのコマンドを格納する	MC07_S_UNIT_COMMAND	56	
	入力PORT構造体 ユニットの入力PORTから読み出された内容を格納する	MC07_S_IN_PORT	58	
	出力PORT構造体 ユニットの出力PORTに書き込むデータ,OR書き込みデータ,AND書き込みデータを格納する	MC07_S_OUT_PORT	61	
	ユニットDRIVE COMMAND･I/O書き込み関数 指定されたユニットに対し、各軸のDATA,COMMAND,各I/O PORTデータの書き込みを一括で行う	MC07_UWDriveIo	63	
	ユニットDRIVE COMMAND書き込み/読み出し関数 指定されたユニットに対し、各軸のDATA,COMMANDを書き込み、読み出しする順の処理を一括で行う	MC07_UWRDrive	65	
	ユニットSTATUS1･I/O読み出し関数 指定されたユニットに対し、各軸のSTATUS1 PORT、I/O PORTの読み出しを一括で行う	MC07_URStatus1Io	66	
	ユニットSTATUS1･パルスカウンタ･I/O読み出し関数 指定されたユニットに対し、各軸のSTATUS1 PORT、パルスカウンタ値、I/O PORTの読み出しを一括で行う	MC07_URStatus1PcntIo	68	

種別	名称	記号	ページ	
	説明		標準編	応用編
ユニットステータス	ユニットI/O PORT書き込み関数	MCO7_UPortOUT	70	
	指定されたユニットに対し、各I/O PORT毎の個別データを書き込む			
	ユニットI/O PORT OR書き込み関数	MCO7_UPortOrOUT	71	
	指定されたユニットに対し、各I/O PORT毎の個別データをOR書き込む			
	ユニットI/O PORT AND書き込み関数	MCO7_UPortAndOUT	72	
	指定されたユニットに対し、各I/O PORT毎の個別データをAND書き込む			
	ユニットI/O PORT読み出し関数	MCO7_UPortIn	73	
	指定されたユニットに対し、各I/O PORTの内容を一括で読み出す			
ドライブ	デバイスオープン関数	MCO7_BOpen	74	
	指定ユニット番号､軸でデバイスオープンし､引数*phDev*の変数にデバイスハンドルを格納する			
	デバイスクローズ関数	MCO7_BClose	75	
	指定されたデバイスをクローズする			
	動作エラークリア関数	MCO7_ClrError	76	
	指定されたデバイスの動作エラーをクリアする			
	DRIVE COMMAND 32ビット書き込み関数	MCO7_LWDrive	79	
	指定デバイスのDRIVE DATA1 PORT､DRIVE DATA2 PORTにデータを書き込んだ後コマンドを書きむ			
	DRIVE COMMAND PORT書き込み関数	MCO7_BWDriveCommand	80	
	指定デバイスのDRIVE COMMAND PORTにコマンドコードを書き込む			
	DRIVE STATUS1 PORT読み出し関数	MCO7_BRStatus1	81	
	指定デバイスのDRIVE STATUS1 PORTを読み出す			
	DRIVE STATUS2 PORT読み出し関数	MCO7_BRStatus2	85	
	指定デバイスのDRIVE STATUS2 PORTを読み出す			
	DRIVE STATUS3 PORT読み出し関数	MCO7_BRStatus3	87	
	指定デバイスのDRIVE STATUS3 PORTを読み出す。			
	DRIVE STATUS4 PORT読み出し関数	MCO7_BRStatus4	88	
	指定デバイスのDRIVE STATUS4 PORTを読み出す。			
	DRIVE STATUS5 PORT読み出し関数	MCO7_BRStatus5	90	
	指定デバイスのDRIVE STATUS5 PORTを読み出す			
	DRIVE STATUSバッファー読み出し関数	MCO7_BRStatusBuf	93	
	指定デバイスのDRIVE STATUS1,STATUS2,STATUS3,STATUS4,STATUS5 PORT,ORIGIN STATUSを読み出す			
	DRIVE COMMAND 32ビット書き込み/読み出し関数	MCO7_LWRDrive	94	
	指定デバイスのDRIVE DATA,COMMAND PORTにデータ､コマンドを書き込み､DRIVE DATA PORTを読み出す			
	DRIVE DATA 32ビット書き込み関数	MCO7_LWDATA		9
	指定デバイスのDRIVE DATA1 PORT､DRIVE DATA2 PORTにデータを書き込む。			
	DRIVE DATA1 PORT書き込み関数	MCO7_BWDriveData1		10
	指定デバイスのDRIVE DATA1 PORTにデータを書き込む。			
	DRIVE DATA2 PORT書き込み関数	MCO7_BWDriveData2		11
	指定デバイスのDRIVE DATA2 PORTにデータを書き込む。			
	DRIVE DATA 32ビット読み出し関数	MCO7_LRDrive		12
	指定デバイスのDRIVE DATA1 PORT、DRIVE DATA2 PORTを一括で読み出す。			
	DRIVE DATA1 PORT読み出し関数	MCO7_BRDriveData1		13
	指定デバイスのDRIVE DATA1 PORTを読み出す。			
	DRIVE DATA2 PORT読み出し関数	MCO7_BRDriveData2		14
	指定デバイスのDRIVE DATA2 PORTを読み出す。			
	データ構造体	MCO7_S_DATA		15
	DRIVE DATA1 PORT、DRIVE DATA2 PORTに書き込むデータを格納する			
	DRIVE COMMANDデータ構造体書き込み関数	MCO7_IWDRIVE		16
	指定デバイスのDRIVE DATA1､2 PORTにデータ構造体の内容を書き込み後､コマンドコードを書き込む			
	DRIVE DATAデータ構造体書き込み関数	MCO7_IWDATA		17
	指定されたデバイスのDRIVE DATA1 PORT、DRIVE DATA2 PORTにデータ構造体の内容を書きむ			
	DRIVE DATA構造体読み出し関数	MCO7_IRDrive		18
	指定されたデバイスのDRIVE DATA1 PORT､DRIVE DATA2 PORTの内容をデータ構造体に読み出す			
	データセット関数	MCO7_SetData		19
	32ビットデータを引数*psData*で示されるデータ構造体に格納する			
	データゲット関数	MCO7_GetData		20
	データ構造体の内容を引数*psData*で示される32ビットデータに変換して返す			

種別	名称 / 説明	記号	ページ 標準編	ページ 応用編
WAIT	READY WAIT関数	MC07_BWaitDriveCommand	95	
	指定デバイスがREADY(STATUS1 BUSY BIT=0)まで待機し、最大待ち時間を超えるとエラー終了する			
	WAIT状態読み出し関数	MC07_BIsWait	96	
	指定デバイスのWAIT状態を返す			
	WAIT中止関数	MC07_BBreakWait	97	
	指定デバイスのREADY WAIT関数またはCOMREG NOT FULL WAIT関数の実行を中止する			
SPEED	SPEED・RATE構造体	MC07_S_SPEED_RATE	98	
	SPEED・RATEセット関数で使用する			
	SPEED・RATEセット関数	MC07_SetSpeedRate	100	
	指定のRESOL No.とSPEED･RATE構造体を元にSPEEDパラメータ設定､加減速時定数(RATE)設定を実行する			
	SPEED・RATE読み出し関数	MC07_ReadSpeedRate	101	
	指定デバイスからSPEEDパラメータ､加減速時定数設定を読み出し､SPEED･RATE構造体に格納する			
ｽﾄﾚｰﾄ補間	POSITION構造体	MC07_S_XY_POSITION	102	
	X･Y座標を指定するときに使用する			
	2軸相対アドレス直線補間ドライブ関数	MC07_IncStrCp	103	
	相対アドレスで指定された目的地まで任意2軸直線補間ドライブを行う			
	2軸相対アドレス円弧補間ドライブ関数	MC07_IncCirCp	105	
	相対アドレスで指定された目的地まで任意2軸円弧補間ドライブを行う			
	メインチップ2軸相対アドレス直線補間ドライブ関数	MC07_McIncStrCp	107	
	相対アドレスで指定された目的地までメインチップ2軸直線補間ドライブを行う			
	メインチップ2軸相対アドレス円弧補間ドライブ関数	MC07_McIncCirCp	109	
	相対アドレスで指定された目的地までメインチップ2軸円弧補間ドライブを行う			
	円の中心点ゲット関数	MC07_GetCirCenterPosition	111	
	円弧の通過点相対アドレス､目的地相対アドレスを元に中心点相対アドレス､回転方向を求める			
	相対アドレス変換関数	MC07_IncFromAbs	112	
	指定された絶対アドレスを相対アドレス(絶対アドレス - 現在位置)に変換する			
	円弧補間短軸PULSE数ゲット関数	MC07_GetCirShortPulse		23
	指定円弧の中心点相対アドレス､目的地相対アドレス､回転方向を元に目的地短軸座標までのパルス数を計算			
ORIGIN	ORIGINドライブパラメータ構造体	MC07_TAG_S_ORG_PARAM	113	
	ORIGINドライブ パラメータ読み出し関数で読み出した内容を格納する			
	ORIGINドライブステータス読み出し関数	MC07_ReadOrgStatus	114	
	ORIGIN STATUSの内容を読み出す			
	ORIGIN SPEC SET関数	MC07_SetOrgSpec	116	
	ORIGINドライブの動作仕様を設定する			
	ORIGIN MARGIN PULSE SET関数	MC07_SetOrgMarginPulse	119	
	ORIGINドライブの機械原点信号検出後のMARGINパルス数を設定する			
	ORIGIN DELAY SET関数	MC07_SetOrgDelay	120	
	ORIGINドライブの各工程後で挿入するDELAYを設定する			
	ORIGIN ERROR PULSE SET関数	MC07_SetOrgErrorPulse	121	
	CONSTANT SCAN工程時および1PULSE送り工程時にエラー判定する各最大パルス数を設定する			
	ORIGIN OFFSET PULSE SET関数	MC07_SetOrgOffsetPulse	122	
	機械原点近傍アドレスのOFFSETパルス数を設定する			
	ORIGIN PRESET PULSE SET関数	MC07_SetOrgPresetPulse	123	
	機械原点検出終了後に実行するORIGINドライブのPRESETパルスを設定する			
	ORIGINドライブパラメータ読み出し関数	MC07_ReadOrgParam	124	
	設定されたORIGINドライブパラメータを読み出し､ORIGINドライブパラメータ構造体に格納する			
	ORIGIN FLAG RESET関数	MC07_ResetOrgFlag	125	
	ORIGIN FLAGをRESETする			
	ORIGINドライブ関数	MC07_Org	126	
	指定された機械原点の型式に従いORIGINドライブを行う			
その他	COMREG NOT FULL WAIT関数	MC07_BWaitComregNotFull		21
	指定デバイスがコマンド予約可能(STATUS1 COMREG FULL BIT=0)まで待機し最大待ち時間超えでエラー終了			
	DRIVE COMMANDバッファー書き込み関数	MC07_LWDriveBuf		22
	コマンドデータ構造体配列に格納さたコマンド,データを指定デバイスのDATA1,2,COMMANDに指定個数書込む。			

種別	名称 / 説明	記号	ページ 標準編	ページ 応用編
I/O	I/O PORTオープン関数 拡張･汎用･制御I/O PORTをオープンし､引数*phPort*の変数にPORTハンドルを格納する	MCO7_BPortOpen	127	
	I/O PORTクローズ関数 指定された拡張･汎用･制御I/O PORTをクローズする	MCO7_BPortClose	129	
	I/O PORT書き込み関数 指定された拡張･汎用･制御・GユニットのI/O PORTにデータを書き込む	MCO7_BPortOut	130	
	I/O PORT OR書き込み関数 指定された拡張･汎用･制御・GユニットのI/O PORTにORデータを書き込む	MCO7_BPortOrOut	131	
	I/O PORT AND書き込み関数 指定された拡張･汎用･制御・GユニットのI/O PORTにANDデータを書き込む	MCO7_BPortAndOut	132	
	I/O PORT 読み出し関数 指定された拡張･汎用･制御・GユニットのI/O PORTのデータを読み出す	MCO7_BPortIn	133	
	I/O PORTラッチエッジ選択書き込み関数 指定された汎用I/O PORTのラッチのエッジを設定する	MCO7_BWLatchEdge	134	
	I/O PORTラッチエッジ選択読み出し関数 指定された汎用I/O PORTのラッチのエッジの設定を読み出す	MCO7_BRLatchEdge	135	
	I/O PORTラッチクリア書き込み関数 指定された汎用I/O PORTのラッチデータをクリアする	MCO7_BWLatchClr	136	
	I/O PORTラッチデータ読み出し関数 指定された汎用I/O PORTのラッチデータを読み出す	MCO7_BRLatchData	137	
数値変換	16ビット符号なし変換関数 指定された16ビット符号付きデータを16ビット符号なしデータに変換する。	MCO7_Unsigned16		24
	16ビット符号付き変換関数 指定された16ビット符号なしデータを16ビット符号つきデータに変換する。	MCO7_signed16		25
	32ビット符号なし変換関数 指定された16ビット符号付きデータを16ビット符号なしデータに変換する。	MCO7_Unsigned32		26
	32ビット符号付き変換関数 指定された16ビット符号なしデータを16ビット符号つきデータに変換する。	MCO7_signed32		27

5-3.ドライブコマンド一覧

(1) 汎用コマンド

種別	コマンド名	コマンドコード	ページ	
	説明		標準編	応用編
エントリ型	NO OPERATION	H'00	160	
	機能なし			
	SPEC INITIALIZE1	H'01	138	
	ドライブパルスの出力仕様の設定			
	SPEC INITIALIZE2	H'02	139	
	CWLM,CCWLM,SSOの設定			
	SPEC INITIALIZE3	H'03	141	
	DRST,DEND/PO,DALMの設定			
	FSPD SET	H'05		35
	ドライブパルス出力の第１パルス目のパルス周期（パルス速度)を設定			
	HIGH SPEED SET	H'06		36
	加減速ドライブの最高速時のパルス速度データ(HSPD)、速度データの速度倍率(RESOL)を設定			
	LOW SPEED SET	H'07		37
	加減速ドライブの加速開始時パルス速度データ(LSPD)、減速終了時パルス速度データ(ELSPD)を設定			
	RATE SET	H'08		38
	加速カーブの変速周期データ(UCYCLE)、減速カーブの変速周期データ(DCYCLE)を設定			
	SCAREA SET	H'09		39
	加速カーブのS字変速領域データ(SUAREA)、減速カーブのS字変速領域データ(SDAREA)を設定			
	DOWN PULSE ADJUST	H'0A		40
	INDEXドライブの自動減速停止動作を開始する減速パルス数のオフセットパルス数を設定			
	JSPD SET	H'0C	144	
	JOGドライブのパルス速度の設定			
	JOG PULSE SET	H'0D	145	
	JOGドライブのパルス数の設定			
	ORIGIN SPEC SET	H'0F		41
	ORIGINドライブの動作仕様を設定			
	+JOG	H'10	146	
	+方向JOGドライブの実行			
	-JOG	H'11	147	
	-方向JOGドライブの実行			
	+SCAN	H'12	148	
	+方向SCANドライブの実行			
	-SCAN	H'13	148	
	-方向SCANドライブの実行			
	INC INDEX	H'14	149	
	相対アドレスINDEXドライブの実行			
	ABS INDEX	H'15	150	
	絶対アドレスINDEXドライブの実行			
	ORIGIN SCAN	H'18		45
	ORIGIN SCANドライブの実行			
	ORIGIN CONSTANT SCAN	H'19		46
	ORIGIN CONSTANT SCANドライブの実行			

種別	コマンド名	コマンドコード	ページ	
	説明		標準編	応用編
ライブラリ	CP_SPEC_SET	H'20		46
	相関2軸1チップのCPPOUT端子から出力するパルスを設定			
	LONG_POSITION_SET	H'22		52
	直線補間ドライブの、長軸の座標アドレスを設定			
	SHORT_POSITION_SET	H'23		53
	直線補間ドライブの、短軸の座標アドレスを設定			
	CIRCULAR_XPOSITION_SET	H'28		61
	円弧の中心点座標を(0,0)とした現在位置のX座標アドレスを設定			
	CIRCULAR_YPOSITION_SET	H'29		62
	円弧の中心点座標を(0,0)とした現在位置のY座標アドレスを設定			
	CIRCULAR_PULSE_SET	H'2A		65
	現在位置のX-Y座標アドレスから目的地の短軸座標までの短軸パルス数を設定			
	MAIN_STRAIGHT_CP	H'30		55
	任意軸、複数軸の直線補間ドライブのときのメイン軸に実行			
	SUB_STRAIGHT_CP	H'31		56
	任意軸、複数軸の直線補間ドライブのときのサブ軸に実行			
	MAIN_XY_STRAIGHT_CP	H'32		54
	メインチップの相関2軸直線補間ドライブの実行			
	MAIN_CIRCULART_CP	H'38		65
	任意軸の円弧補間ドライブのときのメイン軸に実行			
	SUB_CIRCULART_CP	H'39		66
	任意軸の円弧補間ドライブのときのサブ軸に実行			
	MAIN_XY_CIRCULAR_CP	H'3A		64
	メインチップの相関2軸円弧補間ドライブの実行			

(2) 特殊コマンド

種別	コマンド名 / 説明	コマンドコード	ページ 標準編	ページ 応用編
インターポレーション	ADDRESS COUNTER PRESET アドレスカウンタの現在位置設定	H'80	166	
	ADDRESS COUNTER INITIALIZE1 アドレスカウンタの各機能の設定	H'81	161	
	ADDRESS COUNTER INITIALIZE2 アドレスカウンタの各機能の設定	H'82	164	
	ADDRESS COUNTER MAX COUNT SET アドレスカウンタの最大カウント数の設定	H'87		76
	ADRINT COMPARE REGISTER1 SET ADRINTのコンペアレジスタ1の設定	H'88	167	
	ADRINT COMPARE REGISTER2 SET ADRINTのコンペアレジスタ2の設定	H'89	167	
	ADRINT COMPARE REGISTER3 SET ADRINTのコンペアレジスタ3の設定	H'8A	167	
	ADRINT COMP1 ADD DATA SET ADRINTのCOMP1 ADDデータの設定	H'8C	168	
	PULSE COUNTER PRESET パルスカウンタのカウント初期値の設定	H'90	174	
	PULSE COUNTER INITIALIZE1 パルスカウンタの各機能の設定	H'91	169	
	PULSE COUNTER INITIALIZE2 パルスカウンタの各機能の設定	H'92	172	
	PULSE COUNTER MAX COUNT SET パルスカウンタの最大カウント数の設定	H'97		77
	CNTINT COMPARE REGISTER1 SET CNTINTのコンペアレジスタ1の設定	H'98	175	
	CNTINT COMPARE REGISTER2 SET CNTINTのコンペアレジスタ2の設定	H'99	175	
	CNTINT COMPARE REGISTER3 SET CNTINTのコンペアレジスタ3の設定	H'9A	175	
	CNTINT COMP1 ADD DATA SET CNTINTのCOMP1 ADDデータの設定	H'9C	176	
	DFL COUNTER PRESET パルス偏差カウンタのカウント初期値の設定	H'A0	184	
	DFL COUNTER INITIALIZE1 パルス偏差カウンタの各機能の設定	H'A1	177	
	DFL COUNTER INITIALIZE2 パルス偏差カウンタの各機能の設定	H'A2	180	
	DFL COUNTER INITIALIZE3 パルス偏差カウンタの各機能の設定	H'A3	183	
	DFLINT COMPARE REGISTER1 SET DFLINTのコンペアレジスタ1の設定	H'A8	185	
	DFLINT COMPARE REGISTER2 SET DFLINTのコンペアレジスタ2の設定	H'A9	185	
	DFLINT COMPARE REGISTER3 SET DFLINTのコンペアレジスタ3の設定	H'AA	185	
	DFLINT COMP1 ADD DATA SET DFLINTのCOMP1 ADDデータの設定	H'AC	186	

種別	コマンド名	コマンドコード	ページ	
	説明		標準編	応用編
ステッピング系	UDC SPEC SET	H'C0		67
	UP/DOWN/CONSTのドライブCHANGE指令を実行する変更動作点を設定			
	SPEED CHANGE SPEC SET	H'C1		69
	SPEED CHANGE指令を実行する変更動作点を設定			
	INDEX CHANGE SPEC SET	H'C3		72
	INDEX CHANGE指令を実行する変更動作点を設定する			
	UP DRIVE	H'C4		68
	実行中のパルス出力を最高速度まで加速または減速させる			
	DOWN DRIVE	H'C5		68
	実行中のパルス出力を終了速度まで加速または減速させる			
	CONST DRIVE	H'C6		68
	実行中の加速または減速を終了してパルス出力を一定にする			
	SPEED CHANGE	H'C8		70
	実行中のパルス出力を指定ドライブパルス速度まで加速または減速させる			
	RATE CHANGE	H'CA		71
	ドライブCHANGE動作時の変更周期データを指定データに変更			
	INC INDEX CHANGE	H'CC		73
	指定したデータを起動位置を原点とする相対アドレスの停止位置に設定してINC INDEXを行う			
	ABS INDEX CHANGE	H'CD		74
	指定データをアドレスカウンタで管理している絶対アドレスの停止位置に設定してABS INDEXを行う			
	PLS INDEX CHANGE	H'CE		75
	指定データを変更点の検出位置を原点とする相対アドレスの停止位置に設定してINC INDEXを行う			
	ERROR STATUS READ	H'D1	155	
	ERROR STATUSの読み出し			
	MCC SPEED READ	H'D4	157	
	ドライブパルス速度の読み出し			
	MCC SET DATA READ	H'D5	158	
	設定データの読み出し			
	ADDRESS COUNTER READ	H'D8	187	
	アドレスカウンタの読み出し			
	PULSE COUNTER READ	H'D9	187	
	パルスカウンタの読み出し			
	DFL COUNTER READ	H'DA	187	
	パルス偏差カウンタの読み出し			
	ADDRESS LATCH DATA READ	H'DC		80
	アドレスカウンタのラッチデータを読み出す			
	PULSE LATCH DATA READ	H'DD		80
	パルスカウンタのラッチデータを読み出す			
	DFL LATCH DATA READ	H'DE		80
	パルス偏差カウンタのラッチデータを読み出す			
	ERROR STATUS MASK	H'E5	154	
	ERRORに出力するERROR STATUSのマスク			
	COUNT LATCH SPEC SET	H'E8		78
	各種カウンタのカウントデータをラッチするタイミングとクリア機能を設定			

種別	コマンド名	コマンドコード	ページ	
	説明		標準編	応用編
	HARD INITIALIZE1	H'F1		28
	ステータス信号(SOUT信号)に出力する機能を設定			
	HARD INITIALIZE4	H'F4		30
	CWLM, CCWLM信号、DEND/PO信号、DALM信号入力のデジタルフィルタ機能の時定数を設定			
	HARD INITIALIZE5	H'F5		31
	ORG信号、NORG信号、±ZORG信号入力のデジタルフィルタ機能の時定数を設定			
	HARD INITIALIZE6	H'F6		32
	±EA,±EB信号入力のデジタルフィルタ機能の時定数を設定			
	HARD INITIALIZE7	H'F7		33
	CWLM信号、CCWLM信号、FSSTOP信号、DALM信号の入力アクティブ論理を設定			
	SIGNAL OUT	H'FC	152	
	汎用出力信号の操作			
	DRST OUT	H'FD	153	
	DRSTに10ms間のアクティブレベルを出力			
	SLOW STOP	H'FE	151	
	減速停止の実行			
	FAST STOP	H'FF	151	
	即時停止の実行			

本版で改訂された主な箇所

箇　所	内　容
	なし

■ 製品保証

保証期間と保証範囲について

● 納入品の保証期間は、納入後１ヶ年と致します。
● 上記保証期間中に当社の責により故障を生じた場合は、その修理を当社の責任において行います。
(日本国内のみ)
ただし、次に該当する場合は、この保証対象範囲から除外させて頂きます。
(1) お客様の不適当な取り扱い、ならびに使用による場合。
(2) 故障の原因が、当製品以外からの事由による場合。
(3) お客さまの改造、修理による場合。
(4) 製品出荷当時の科学・技術水準では予見が不可能だった事由による場合。
(5) その他、天災、災害等、当社の責にない場合。
(注1)ここでいう保証は､納入品単体の保証を意味するもので納入品の故障により誘発される損害はご容赦頂きます。
(注2)当社において修理済みの製品に関しましては、保証外とさせて頂きます。

技術相談のお問い合わせ

TEL.(042)664-5382 FAX.(042)666-5664
E-mail s-support@melec-inc.com

販売に関するお問い合わせ

TEL.(042)664-5384 FAX.(042)666-2031

株式会社メレック 制御機器営業部
〒193-0834 東京都八王子市東浅川町516-10 URL:http://www.melec-inc.com

記載内容は、製品改良のため予告なく変更することがありますのでご了承ください。 c1503